

CICS Transaction Server for z/OS
Version 4 Release 2



System Programming Reference

CICS Transaction Server for z/OS
Version 4 Release 2



System Programming Reference

Note

Before using this information and the product it supports, read the information in "Notices" on page 865.

This edition applies to Version 4 Release 2 of CICS Transaction Server for z/OS (product number 5655-S97) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 1977, 2012.**

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

Contents

Preface	vii
What this book is about	vii
Who should read this book	vii
What you need to know to understand this book	vii
How to use this book	vii
Terminology	vii

Changes in CICS Transaction Server for z/OS, Version 4 Release 2	ix
---	-----------

Chapter 1. Introduction to system programming commands **1**

Command format	2
CICS syntax notation	3
Summary of format rules	5
Argument values	6
Data-areas and data-values	6
Pointer arguments	8
CICS-value data areas (CVDAs)	8
CVDA examples	9
Data types	10
COBOL argument values	11
C and C++ argument values	11
PL/I argument values	12
Assembly language argument values	13
Argument lengths	14
Null values	14
Exception conditions	14
RESP and RESP2 options	15
Security checking	16
Inquiry commands	19
Browsing resource definitions	20
Starting a browse	20
Retrieving the next resource	21
Ending the browse	22
Browse example	22
Rules for browsing	22
Exception conditions for browsing	23
SET commands	23
Creating resource definitions	24
The ATTRIBUTES option	25
Discarding resource definitions	27
Exit-related commands	28
Defining exits	28
Exit names	29
CICS threadsafe commands in the SPI	29

Chapter 2. System commands **31**

ACQUIRE TERMINAL	31
COLLECT STATISTICS	33
Collecting statistics for Java programs that run in a JVM	41
CREATE ATOMSERVICE	42
CREATE BUNDLE	44

CREATE CONNECTION	45
CREATE CORBASERVER	49
CREATE DB2CONN	51
CREATE DB2ENTRY	54
CREATE DB2TRAN	56
CREATE DJAR	58
CREATE DOCTEMPLATE	60
CREATE ENQMODEL	61
CREATE FILE	63
CREATE IPCONN	66
CREATE JOURNALMODEL	70
CREATE JVMSERVER	72
CREATE LIBRARY	73
CREATE LSRPOOL	77
CREATE MAPSET	80
CREATE MQCONN	81
CREATE PARTITIONSET	83
CREATE PARTNER	85
CREATE PIPELINE	87
CREATE PROCESSTYPE	88
CREATE PROFILE	90
CREATE PROGRAM	93
CREATE REQUESTMODEL	95
CREATE SESSIONS	97
CREATE TCPIPSERVICE	100
CREATE TDQUEUE	103
CREATE TERMINAL	106
CREATE TRANCLASS	109
CREATE TRANSACTION	111
CREATE TSMODEL	114
CREATE TYPETERM	116
CREATE URIMAP	120
CREATE WEBSERVICE	123
CSD ADD	124
CSD ALTER	126
CSD APPEND	129
CSD COPY	131
CSD DEFINE	133
CSD DELETE	137
CSD DISCONNECT	139
CSD ENDBRGROUP	139
CSD ENDBRLIST	140
CSD ENDBRRSRCE	141
CSD GETNEXTGROUP	142
CSD GETNEXTLIST	143
CSD GETNEXTSRCE	143
CSD INQUIREGROUP	146
CSD INQUIRELIST	147
CSD INQUIRERSRCE	147
CSD INSTALL	150
CSD LOCK	153
CSD REMOVE	154
CSD RENAME	156
CSD STARTBRGROUP	159
CSD STARTBRLIST	160
CSD STARTBRRSRCE	161

CSD UNLOCK	162	INQUIRE DB2TRAN	291
CSD USERDEFINE	163	INQUIRE DELETSHIPED	294
DISABLE PROGRAM	166	INQUIRE DISPATCHER	295
DISCARD ATOMSERVICE	170	INQUIRE DJAR	298
DISCARD AUTINSTMODEL	171	INQUIRE DOCTEMPLATE	301
DISCARD BUNDLE	172	INQUIRE DSNNAME	306
DISCARD CONNECTION	172	INQUIRE DUMPDS	312
DISCARD CORBASERVER	174	INQUIRE ENQ	313
DISCARD DB2CONN	175	INQUIRE ENQMODEL	314
DISCARD DB2ENTRY	175	INQUIRE EPADAPTER	317
DISCARD DB2TRAN	176	INQUIRE EVENTBINDING	322
DISCARD DJAR	177	INQUIRE EVENTPROCESS	324
DISCARD DOCTEMPLATE	178	INQUIRE EXCI	325
DISCARD ENQMODEL	179	INQUIRE EXITPROGRAM	326
DISCARD FILE	179	INQUIRE FILE	332
DISCARD IPCONN	181	INQUIRE HOST	347
DISCARD JOURNALMODEL	181	INQUIRE IPCONN	348
DISCARD JOURNALNAME	182	INQUIRE IPFACILITY	357
DISCARD JVMSERVER	183	INQUIRE IRC	358
DISCARD LIBRARY	184	INQUIRE JOURNALMODEL	359
DISCARD MQCONN	185	INQUIRE JOURNALNAME	362
DISCARD PARTNER	186	INQUIRE JOURNALNUM	364
DISCARD PIPELINE	187	INQUIRE JVM	364
DISCARD PROCESSTYPE	188	INQUIRE JVMPPOOL	367
DISCARD PROFILE	188	INQUIRE JVMPROFILE	368
DISCARD PROGRAM	189	INQUIRE JVMSERVER	370
DISCARD REQUESTMODEL	190	INQUIRE LIBRARY	373
DISCARD TCPIPSERVICE	191	INQUIRE MODENAME	378
DISCARD TDQUEUE	192	INQUIRE MONITOR	380
DISCARD TERMINAL	193	INQUIRE MQCONN	384
DISCARD TRANCLASS	194	INQUIRE MQINI	388
DISCARD TRANSACTION	195	INQUIRE MVSTCB	390
DISCARD TSMODEL	196	INQUIRE NETNAME	392
DISCARD URIMAP	197	INQUIRE OSGIBUNDLE	393
DISCARD WEBSERVICE	198	INQUIRE OSGISERVICE	397
ENABLE PROGRAM	199	INQUIRE PARTNER	398
Enabling global user exits	204	INQUIRE PIPELINE	400
Enabling task-related user exits	205	INQUIRE PROCESSTYPE	404
EXTRACT EXIT	205	INQUIRE PROFILE	407
EXTRACT STATISTICS	207	INQUIRE PROGRAM	410
INQUIRE ASSOCIATION	212	INQUIRE REQID	421
INQUIRE ASSOCIATION LIST	222	INQUIRE REQUESTMODEL	426
INQUIRE ATOMSERVICE	225	INQUIRE RRMS	429
INQUIRE AUTINSTMODEL	229	INQUIRE STATISTICS	429
INQUIRE AUTOINSTALL	230	INQUIRE STORAGE	432
INQUIRE BEAN	232	INQUIRE STREAMNAME	434
INQUIRE BRFACILITY	233	INQUIRE SUBPOOL	436
INQUIRE BUNDLE	235	INQUIRE SYSDUMPCODE	437
INQUIRE BUNDLEPART	239	INQUIRE SYSTEM	440
INQUIRE CAPDATAPRED	241	INQUIRE TASK	452
INQUIRE CAPINFOSRCE	244	INQUIRE TASK LIST	463
INQUIRE CAPOPTPRED	246	INQUIRE TCLASS	464
INQUIRE CAPTURESPEC	249	INQUIRE TCPIP	465
INQUIRE CFDTPOOL	254	INQUIRE TCPIPSERVICE	467
INQUIRE CLASSCACHE	256	INQUIRE TDQUEUE	475
INQUIRE CONNECTION	258	INQUIRE TEMPSTORAGE	485
Remote connections	268	INQUIRE TERMINAL	485
Example of remote definitions	269	Remote terminals	504
INQUIRE CORBASERVER	270	INQUIRE TRACEDEST	505
INQUIRE DB2CONN	275	INQUIRE TRACEFLAG	507
INQUIRE DB2ENTRY	285	INQUIRE TRACETYPE	509

INQUIRE TRANCLASS	511
INQUIRE TRANDUMPCODE	514
INQUIRE TRANSACTION	517
INQUIRE TSMODEL	527
INQUIRE TSPOOL	530
INQUIRE TSQUEUE / TSQNAME	531
INQUIRE UOW	535
INQUIRE UOWDSNFAIL	539
INQUIRE UOWENQ	544
INQUIRE UOWLINK	550
INQUIRE URIMAP	554
INQUIRE VOLUME	561
INQUIRE VTAM	561
INQUIRE WEB	564
INQUIRE WEBSERVICE	565
INQUIRE WORKREQUEST	571
INQUIRE XMLTRANSFORM	575
PERFORM CLASSCACHE	579
PERFORM CORBASERVER	582
PERFORM DELETSHIPED	585
PERFORM DJAR	586
PERFORM DUMP	587
PERFORM ENDAFFINITY	589
PERFORM JVMPOOL	591
PERFORM PIPELINE	593
PERFORM RESETTIME	594
PERFORM SECURITY REBUILD	595
PERFORM SHUTDOWN	596
PERFORM STATISTICS RECORD	598
RESYNC ENTRYNAME	605
SET ATOMSERVICE	608
SET AUTOINSTALL	609
SET BRFACILITY	610
SET BUNDLE	611
SET CLASSCACHE	612
SET CONNECTION	614
SET CORBASERVER	622
SET DB2CONN	624
SET DB2ENTRY	634
SET DB2TRAN	640
SET DELETSHIPED	641
SET DISPATCHER	643
SET DOCTEMPLATE	646
SET DSNAME	647
SET DUMPDS	656
SET ENQMODEL	658
SET EPADAPTER	659
SET EVENTBINDING	660
SET EVENTPROCESS	661
SET FILE	662
SET HOST	675
SET IPCONN	676
SET IRC	681
SET JOURNALNAME	683
SET JOURNALNUM	685
SET JVMPOOL	685
SET JMVSERVER	686
SET LIBRARY	689
SET MODENAME	691
SET MONITOR	693
SET MQCONN	698

SET NETNAME	701
SET PIPELINE	702
SET PROCESSTYPE	703
SET PROGRAM	704
SET STATISTICS	708
SET SYSDUMPCODE	712
SET SYSTEM	715
SET TASK	721
SET TCLASS	723
SET TCPIP	724
SET TCPIPSERVICE	726
SET TDQUEUE	728
SET TEMPSTORAGE	732
SET TERMINAL	733
SET TRACEDEST	742
SET TRACEFLAG	746
SET TRACETYPE	748
SET TRANCLASS	751
SET TRANDUMPCODE	752
SET TRANSACTION	756
SET TSQUEUE / TSQNAME	759
SET UOW	761
SET UOWLINK	762
SET URIMAP	763
SET VOLUME	765
SET VTAM	765
SET WEB	768
SET WEBSERVICE	769
SET WORKREQUEST	770
SET XMLTRANSFORM	772

Appendix A. CICS-value data areas	
used by all commands	775
CVDAs and numeric values in alphabetic sequence	775
CVDAs and numeric values in numeric sequence	799
CVDA values for the DEVICE option	822

Appendix B. EXEC interface block	
(EIB) response and function codes	825
Response codes of EXEC CICS commands	825
EXEC CICS CSD, DISCARD, INQUIRE,	
PERFORM, and SET commands	825
EXEC CICS DISABLE, ENABLE, and EXTRACT	
EXIT commands	826
Function codes of EXEC CICS commands	826

Appendix C. RESP2 values for EXEC	
CICS CREATE and EXEC CICS CSD	
commands	837

Appendix D. Threadsafe SPI	
commands	863

Notices	865
Trademarks	866

Bibliography	867
CICS books for CICS Transaction Server for z/OS	867

CICSplex SM books for CICS Transaction Server
for z/OS 868
Other CICS publications 868
Other IBM publications 868

Accessibility 871
Index 873

Preface

What this book is about

This book describes the CICS[®] system programming interface. It contains reference information needed to prepare COBOL, C, C++, PL/I, and assembler-language application programs, using CICS commands, to be executed under CICS.

Who should read this book

This book is for system programmers who are writing applications to be invoked as transactions for administering the running CICS system.

What you need to know to understand this book

It is assumed that you are an experienced system programmer and that you are familiar with the effects of the CICS-supplied transactions.

You must be able to write application programs, and understand the contents of the CICS application programming books (that is, the *CICS Application Programming Reference* and the *CICS Application Programming Guide*). Anything that is already documented in those two books is not duplicated here, so you may need to refer to them occasionally.

How to use this book

This book contains two major sections. The first section is an introduction that describes the common features and the overall purpose of the system programming interface commands. The second section is a description of each of the commands, in alphabetic order.

Terminology

- ‘CICS’ refers to IBM[®] CICS Transaction Server for z/OS[®], Version 4 Release 2 .
- ‘VTAM’ refers to IBM ACF/VTAM
- The term ‘SP’ indicates those commands that require the special translator option ‘SP’. It also indicates those commands that are subject to command security checking. The SP commands are all the INQUIRE, SET, COLLECT, PERFORM, CREATE, and DISCARD commands, together with the DISABLE PROGRAM, ENABLE PROGRAM, EXTRACT EXIT, EXTRACT STATISTICS, and RESYNC ENTRYNAME commands and some of the front end programming interface (FEPI) commands.
- MB equals 1 048 576 bytes.
- KB equals 1024 bytes.

Changes in CICS Transaction Server for z/OS, Version 4 Release 2

For information about changes that have been made in this release, please refer to *What's New* in the information center, or the following publications:

- *CICS Transaction Server for z/OS What's New*
- *CICS Transaction Server for z/OS Upgrading from CICS TS Version 4.1*
- *CICS Transaction Server for z/OS Upgrading from CICS TS Version 3.2*
- *CICS Transaction Server for z/OS Upgrading from CICS TS Version 3.1*

Any technical changes that are made to the text after release are indicated by a vertical bar (|) to the left of each new or changed line of information.

Chapter 1. Introduction to system programming commands

The CICS system programming interface (SPI) commands are for managing the CICS system and its resources, in contrast to the application programming interface (API) commands, with which you implement end-user applications.

The API is described in CICS API commands in CICS Application Programming; the Application programming overview in CICS Application Programming contains general information that applies to both groups of commands.

SPI commands either retrieve information about the system and its resources, or modify them. They fall into three broad categories:

- Commands that retrieve information about a CICS resource or system element:
 - The INQUIRE commands
 - COLLECT STATISTICS
 - EXTRACT STATISTICS
- Commands that modify the status or definition of the system or a resource, or invoke a system process:
 - The SET commands
 - The CREATE commands
 - The DISCARD commands
 - The PERFORM commands
 - ACQUIRE TERMINAL
- Commands that modify or expand system execution by means of exits:
 - DISABLE PROGRAM
 - ENABLE PROGRAM
 - EXTRACT EXIT
 - RESYNC ENTRYNAME

Together, these commands provide you with a command-level equivalent to the function of the master terminal transaction (CEMT) and the trace control transaction (CETR), and as an alternative to the CEDA transaction for defining resources. This means that you can write transactions for administering the running CICS system. You could, for example, provide some functions of the master terminal command for a group of users without giving them authority to use CEMT.

System programming commands are supported in the same way as application programming commands. They can be used in programs written in any CICS-supported language, and they are recognized by the command interpreter (CECI), the execution diagnostic facility (EDF), and the CICS translator.

However, there are some differences between SPI and API commands:

- You cannot function ship SPI commands by naming a remote resource or, generally, by specifying the SYSID option. They are executed in the CICS region in which the issuing program is running. If the command specifies a remote resource (one owned by another region), CICS uses the local (partial) definition to process the request. Consequently, if you want to use or change a resource definition in a remote region, you must cause your SPI command to be executed in that region, either by transaction routing or by distributed program link. Shared temporary storage queues are an exception.

- Additional security checking is available for SPI commands, as explained in The format of SPI commands.
- Programs containing SPI commands must be translated with the SP translator option, as explained in “Security checking” on page 16.

There are also special considerations that apply to certain groups of commands. They are described in “Inquiry commands” on page 19.

Command format

You write SPI commands in the same way as API commands. SPI commands begin with the words **EXECUTE CICS**, usually abbreviated **EXEC CICS**, followed by the command name, a verb, or verb-and-option combination.

For example, an SPI command could look as follows:

- INQUIRE FILE
- PERFORM SHUTDOWN
- SET SYSTEM

Options that indicate details of what you want to do follow the command name. The order of the options is unimportant except when the first one is part of the command name (the FILE in INQUIRE FILE, for example).

SPI commands are translated into the language of the program by the same CICS translator that converts API commands, and you can mix the two categories of commands in the same program. However, you must specify the translator option SP when SPI commands are present, or the translator will not recognize them. This feature allows an installation to limit use of the SPI at compile time. Other security features restrict its use at execution time; these are described in “Security checking” on page 16.

The EXEC CICS that begins a command tells the translator when to begin translating. In high-level languages, you must also tell the translator when to stop, by putting a terminator at the end of the command. In COBOL, the terminator is **END-EXEC**. In C and PL/I, it is a semi-colon. You do not need one in assembler, because the translator assumes that the command ends on the current line unless a continuation character is present. So a command that looks like this in assembler:

```
EXEC CICS SET FILE(TAXPGM) OPEN
```

becomes

```
EXEC CICS SET FILE(TAXPGM) OPEN END-EXEC
```

in COBOL, and

```
EXEC CICS SET FILE(TAXPGM) OPEN;
```

in C or PL/I.

For more information about translating the commands, see the *CICS Application Programming Guide*

CICS syntax notation

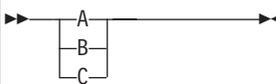
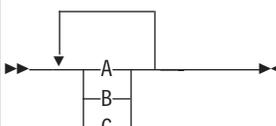
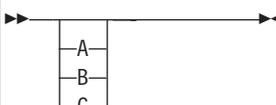
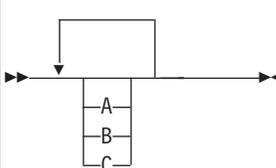
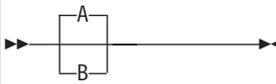
The syntax for each command is presented in the form of a diagram. The diagram tells you what you can put between the EXEC CICS that begins a command and the terminator that ends it. It summarizes what you can do with the particular command, and indicates relationships between different options and, sometimes, different values of an option.

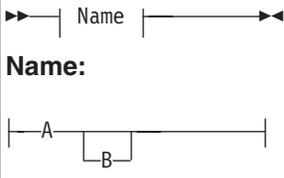
The diagrams and some of the examples omit the initial EXEC CICS and the language-dependent terminator, even though you must use them in your code. The diagrams also omit options that you can use in any command:

- NOHANDLE
- RESP
- RESP2
- SYSEIB

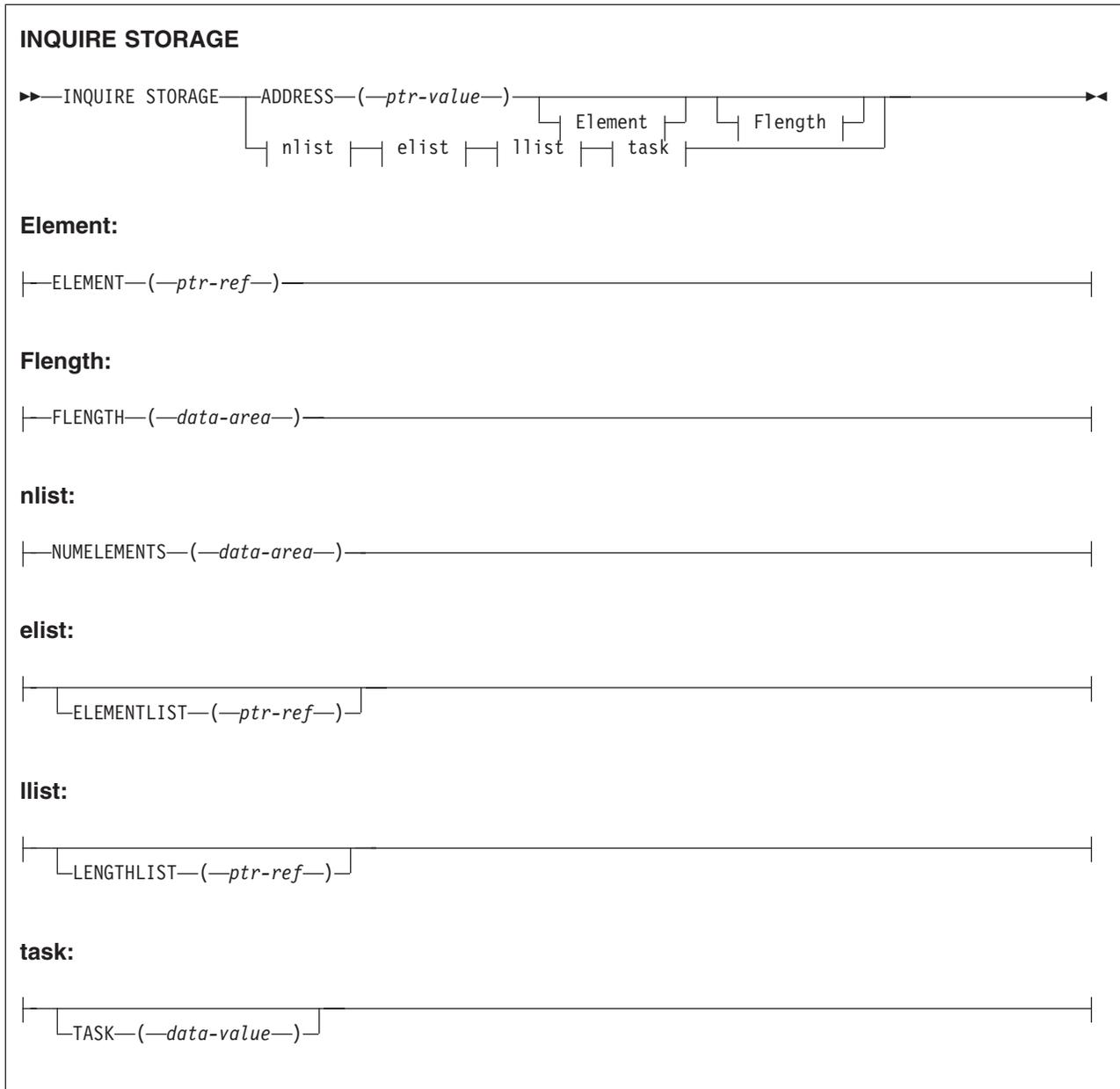
These have the same meaning in SPI commands as in API commands. (See the *CICS Application Programming Guide* for basic information about these options, and "Exception conditions" on page 14 for additional SPI details.)

You read the diagram by following the arrows from left to right, using these conventions:

Symbol	Action
	A set of alternatives - one of which you must code.
	A set of alternatives - one of which you must code. You can code more than one of them, in any sequence.
	A set of alternatives - one of which you might code.
	A set of alternatives - any number, including none, of which you might code once, in any sequence.
	Alternatives where A is the default.

Symbol	Action
 <p>Name:</p>	Use with the named section in place of its name.
Punctuation and uppercase characters	Code exactly as shown.
Lowercase characters	Code your own text, as appropriate; for example, name.

Below is an example. It indicates that INQUIRE STORAGE requires you to specify either the ADDRESS option or the NUMELEMENTS option (but not both). If, and only if, you choose ADDRESS, you can specify ELEMENT, FLENGTH, both, or neither. If you choose NUMELEMENTS, you can specify DSANAME, ELEMENTLIST, LENGTHLIST, or TASK in any combination (including none).



Conditions: NOTAUTH, TASKIDERR

A list of the exception conditions that can occur on the command appears at the bottom of the diagram. In this case, the possibilities are the NOTAUTH and TASKIDERR conditions.

Summary of format rules

Here is a summary of the format rules for coding CICS commands.

- Follow the conventions of the language in which you are coding for general format (the column in which the command starts, the columns available to it, embedded comments, embedded blanks, and so on).

Note: The translator is not sensitive to blanks between option names and option values or the parentheses that separate them, so you can use them or not, as you want, even in assembler.

- Start your command with **EXEC CICS** or **EXECUTE CICS** and end it with the terminator required by the program language (see “Command format” on page 2).
- If a command does not fit on a line, or you want to break it into multiple lines, use the conventions of the language. In assembler, use a continuation character on all but the last line.
- Select the options you want to use from the syntax diagram, observing the rules expressed in the diagram and the option text for required options and consistent combinations.
- Code punctuation and upper case letters as shown in the diagram (you can use mixed case or lowercase for keywords shown in uppercase if you prefer).
- Substitute your own text for lowercase letters, following the conventions of the language in which you are coding.

Argument values

The data associated with an option is called its **argument**.

Five different types appear in the syntax diagrams:

- data-area
- data-value
- ptr-ref (for pointer-reference)
- ptr-value (pointer-value)
- cvda (CICS-value data area)

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. For example, in the command:

```
EXEC CICS INQUIRE PROGRAM (TAXPGM)
        USECOUNT (UCNT) END-EXEC
```

PROGRAM is a sender option and **TAXPGM** is a data-value; it tells CICS where to find the name of the program you are inquiring about. **USECOUNT** is a receiver option, and **UCNT** is a data-area; CICS returns the information you requested (the use count for this program) there.

In general, you can use any area (variable) for a data-area, provided that:

- The data type (format) is correct. The area must be long enough and, in high-level languages, the associated variable must be defined to have the correct length and internal representation. The data types that CICS uses are discussed in “Data types” on page 10.
- The program logic allows the value to be changed (CICS stores into data-areas).
- CICS re-entrancy rules allow the value to be changed. CICS loads only one copy of any given program, no matter how many tasks are using it. To prevent tasks executing the same program from interfering with one another, CICS keeps a separate copy of program areas that may change during execution (sometimes called “working storage”) for each task. This means that any area that may be modified, including data-area arguments to CICS commands, must reside either

in such an area of the program or in storage outside the program which the application design allows the program to modify.

Some of this storage is allocated automatically; this category includes the WORKING-STORAGE section in COBOL programs, AUTOMATIC storage in PL/I and C/370, and areas appended to the DFHEISTG DSECT in assembler. It can also be allocated explicitly with a CICS GETMAIN command or a language facility such as a PL/I ALLOCATE statement, in this or a preceding program. This category includes the LINKAGE section in COBOL, BASED and CONTROLLED storage in PL/I, and other DSECTs in assembler. See the *CICS Application Programming Guide* for more detail about CICS re-entrancy rules.

- The program that issues the command has *write* access to the area. CICS changes the content of data-areas and, therefore, you cannot use storage that you are not allowed to change.

Write access is affected by the storage protection key in which the program is running, and by the transaction isolation status of its task. See the discussion of these subjects in the *CICS Application Programming Guide* and the *CICS System Definition Guide*, and the TRANISOLATE option of a TRANSACTION definition in the *CICS Resource Definition Guide*.

- The MVS restrictions on addressing mode that apply to all CICS commands are observed. These are enforced automatically in high-level languages but, in assembler, the program must be in primary addressing mode, and the primary address space must be the home address space. All arguments for options must reside in the primary address space.

Note: 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.

Any area that can be used for a data-area can also be used for a data-value. In addition, you can use areas not allowed for data-areas, because CICS never changes a data-value. In particular, you can use:

- Constants, including literals. In the example above, for instance, you could use a literal instead of a variable for the program name:

```
EXEC CICS SET TDQUEUE ('TAX')
        TRIGGERLEVEL(1) END-EXEC
```

When you use a numeric literal in a command, the translator ensures a constant of the correct type and length, provided the literal is capable of being converted to such a constant, as in TRIGGERLEVEL above. In COBOL and assembler, the translator also ensures character literals of the correct length, padding with blanks if the literal is shorter than the length the argument requires. In C/370 and PL/I, however, you must do this yourself:

```
EXEC CICS SET TDQUEUE ('TAX ')
        TRIGGERLEVEL(1);
```

- Other program areas not in “working storage”, such as static storage in PL/I.
- Areas to which your program has read but not write access (the link-pack area, for example).

Note: Sometimes an option is used both to send and receive information, although this usage occurs more often in API than SPI commands. When it does, the argument must be a data-area, because CICS stores into it.

Pointer arguments

A pointer-reference, abbreviated to ptr-ref in the diagrams, is a special case of a data-area. It also is a receiver field, but CICS uses it to return a pointer to the data requested, rather than the data itself; that is, CICS stores the location (address) of the data in the argument you provide.

A pointer-value (abbreviated ptr-value) is the pointer counterpart of a data-value; that is, you send information to CICS in a pointer-value, but you provide the address of the data (a pointer to it), rather than the data itself.

The rules listed for data-areas therefore apply to pointer-references, and those for data-values to pointer-values. Each language provides a type definition for pointers, and facilities for expressing address literals that can be used for pointer-values; internally, pointers are stored in fullword binary form. See command the *CICS Application Programming Reference* for more information about the distinction between data and pointers.

CICS-value data areas (CVDAs)

A CVDA (CICS-value data area) is an argument to which CICS has assigned a specific and limited set of meaningful values. These values are named, both to make them intuitive and easy to remember and to keep the interface between user programs and CICS symbolic, so that version and platform changes do not require program modifications.

Some CVDAs send information to CICS. A sender CVDA is a special case of a data-value, and the rules for data-values apply. Others return information from CICS, and you must use the rules for data-areas. If there is any question about the direction in which the information is flowing, you can tell from the verb used in the option description. Specifies means that you are sending information to CICS (that is, data-value rules apply); returns indicates that CICS will return information in the argument (data-area rules apply).

CICS provides the code that converts CVDA value names to the corresponding numeric representations. Internally, CVDAs are stored as fullword binary numbers, and you must always provide a fullword binary area for options that receive CVDA values.

One way to send a CVDA value is to name the appropriate value (the name of the option is implied in the name of the value). For example:

```
EXEC CICS SET PROGRAM (TAXPGM)
        DPLSUBSET END-EXEC
```

sets the EXECUTIONSET option value to DPLSUBSET. EXECUTIONSET determines the set of commands which the program is allowed to use. It has two possible values: DPLSUBSET, which restricts a program to the commands allowed in a program invoked by a distributed program link, and FULLAPI, which does not restrict the command set.

The alternative is to use the CICS-provided DFHVALUE function, which relates the internal representation to the value name. For example, this code is equivalent to the COBOL statement above:

```
MOVE DFHVALUE(DPLSUBSET) TO TAXAPI.
EXEC CICS SET PROGRAM (TAXPGM)
        EXECUTIONSET(TAXAPI) END-EXEC.
```

This technique is easier to use when program logic is complex.

You also use DFHVALUE when your program needs to interpret a value returned as a CVDA. For example, if you needed to perform logic based on the EXECUTIONSET value, you would write something like this:

```
EXEC CICS INQUIRE PROGRAM (TAXPGM)
           EXECUTIONSET (TAXAPI) END-EXEC.
IF TAXAPI = DFHVALUE(FULLAPI) PERFORM STND-INIT
ELSE PERFORM REMOTE-INIT.
```

CICS-value data areas used by all commands in CICS System Programming Reference lists all of the CVDA value names with corresponding numeric values. These are for reference only, however; you should use value names and DFHVALUE in your code, to keep it version- and platform-independent.

CVDA examples

Here are examples in all the CICS-supported languages that show the use of CVDA's and the DFHVALUE function.

In each case, the code does the following:

- Tests whether the file named PAYROLL is closed.
- If so, changes the UPDATE and DELETE option values for the file to UPDATABLE and NOTDELETABLE respectively (so that records can be updated and read, but not deleted). Note that the UPDATE option is set by using the DFHVALUE function, and that the DELETE option is set by specifying the value name. These methods are equivalent; either could have been done either way.

The absence of other options indicates that those values are to remain unchanged. This information could also have been expressed by specifying the options with null values, as explained in "SET commands" on page 23.

- Returns to CICS.

Only the code and definitions related to this part of each program are shown.

COBOL version

```
WORKING-STORAGE SECTION.
01 FILE-STATUS-INFO.
   02 UOPST          PIC S9(8) COMP.
   02 UUPD          PIC S9(8) COMP.
   02 INFILE        PIC X(8).
. . .
CICS-REQUESTS.
  MOVE 'PAYROLL ' TO INFILE.
  EXEC CICS INQUIRE FILE(INFILE)
           OPENSTATUS(UOPST) END-EXEC.
  IF UOPST = DFHVALUE(CLOSED)
    MOVE DFHVALUE(UPDATABLE) TO UUPD
    EXEC CICS SET FILE(INFILE)
           UPDATE(UUPD)
           NOTDELETABLE END-EXEC.
  EXEC CICS RETURN.
```

C version

```

#define INFILE    "PAYROLL "
main()
{
    long int uopst,    /* OPENSTATUS value */
    long int uupd;    /* UPDATE value */
    . . .
    EXEC CICS ADDRESS EIB(dfheiptr);
    EXEC CICS INQUIRE FILE(INFILE)
                OPENSTATUS(uopst);
    if( uopst == DFHVALUE(CLOSED) )
    { uupd = DFHVALUE(UPDATABLE);
      EXEC CICS SET FILE(INFILE)
                UPDATE(uupd)
                NOTDELETABLE; }
    EXEC CICS RETURN;
}

```

PL/I version

```

DCL (UOPST,UUPD) FIXED BIN(31), /*OPEN,UPD STATUS*/
    INFILE CHAR(8);           /*FILE NAME */
    . . .
    INFILE='PAYROLL ';
    EXEC CICS INQUIRE FILE(INFILE)
                OPENSTATUS(UOPST):
    IF UOPST = DFHVALUE(CLOSED) THEN DO;
        UUPD = DFHVALUE(UPDATABLE);
        EXEC CICS SET FILE(INFILE)
                UPDATE(UUPD)
                NOTDELETABLE; END;
    EXEC CICS RETURN;

```

Assembler-language version

```

                DFHEISTG
UOPST          DS      F          *OPEN STATUS
UUPD           DS      F          *UPDATE STATUS
INFILE         DS      CL8       *FILE NAME
                . . .
                MVC INFILE,=CL8'PAYROLL '
                EXEC CICS INQUIRE FILE(INFILE)          X
                    OPENSTATUS(UOPST)
                CLC  UOPST,DFHVALUE(CLOSED)
                BNE  OPENLAB
                MVC  UUPD,DFHVALUE(UPDATABLE)
                EXEC CICS SET FILE(INFILE)              X
                    UPDATE(UUPD)                       X
                    NOTDELETABLE
OPENLAB        EXEC CICS RETURN

```

Data types

For most arguments, CICS uses one of five data types. The first four data types are all used for numeric data, but they differ in length and internal format; the last is for text. The names used in this information are those used in assembler language.

The five data types are as follows:

- Doubleword binary (eight bytes)

- Fullword binary (four bytes)
- Halfword binary (two bytes)
- Packed decimal (variable number of bytes)
- Character string (variable number of bytes)

Data-areas and data-values might require any of these formats. The option text tells you which one to use. CVDA's are always fullword binary. Pointers are also stored in this form, although you generally define them explicitly as pointers or addresses. There are a few exceptions to these types, including the component identifier arguments in the INQUIRE and SET TRACETYPE commands, which are bit strings, options where the user determines the data format, and options for which CICS requires a specific structure. These exceptions are rare in the SPI, however, and are always noted in the option description text.

The data types are the same regardless of the language of the program issuing the command. However, the way you define data of a particular type varies with the language. The rules are summarized in the language sections that follow, but there are other considerations unique to each language. You should refer to the relevant language manual for information, although you can find some language-specific information in the Programming languages and Language Environment in CICS Application Programming

COBOL argument values

In COBOL, you can use any data name of the correct data type for any argument. For a *data-value*, you can also use a constant that can be converted to the correct type.

The ADDRESS special register can be used for both *pointer-references* and *pointer-values*, and the LENGTH special register can be used for length arguments that take a *data-value*. The table that follows indicates how to define the correct data type.

Data type	COBOL definition
Halfword binary	PIC S9(4) COMP
Fullword binary (including CVDA)	PIC S9(8) COMP
Doubleword binary	PIC S9(18) COMP
Pointer	USAGE IS POINTER
Character string (<i>n</i> characters long)	PIC X(<i>n</i>)
UTF-8 character string (<i>n</i> bytes long)	PIC X(<i>n</i>)
Packed decimal (<i>n</i> decimal digits)	PIC S9(<i>n</i>) COMP-3

C and C++ argument values

In C and C++, you can use any data reference of the correct data type for a *data-area*, *data-value*, or CVDA, provided the reference is to contiguous storage. In addition, for a *data-value*, you can use any C expression that can be converted to the correct data type.

The following table shows how to define the correct data type:

Data type	C definition
Halfword binary	short int

Data type	C definition
Fullword binary (including CVDA)	long int
Doubleword binary	long long int
Character string (<i>n</i> characters long)	unsigned char[<i>n</i>]
UTF-8 character string (<i>n</i> bytes long)	char[<i>n</i>]
Packed decimal	Not used - see note
Note: Packed decimal arguments are not supported in C and C++. Whenever there is an option that takes such an argument, there are other options that convey or return the same information in a format supported by C and C++.	

Pointer-reference and pointer-value arguments can be any C or C++ *pointer-reference*, and *pointer-values* can also be any C or C++ expression that can be converted to an address.

CICS calling sequences pass arguments by reference (the MVS convention), rather than by value (the C convention). Ordinarily, the translator makes the necessary adjustments, but there are some situations in which you need to prefix your argument with an ampersand (&). See the C discussion in the *CICS Application Programming Guide* for details on arguments and other aspects of writing CICS programs in C and C++.

PL/I argument values

In PL/I, an argument can be any PL/I data reference of the correct data type, provided the reference is to connected storage. In addition, a *data-value*, a *pointer-value*, or a sender CVDA can be any PL/I expression that can be converted to the required type, including one containing built-in functions like ADDR or LENGTH.

The following table shows how to define the correct data type:

Data type	PL/I definition
Halfword binary	FIXED BIN(15)
Fullword binary (including CVDA)	FIXED BIN(31)
Doubleword binary	FIXED BIN(63)
Pointer	POINTER
Character string (<i>n</i> characters long)	CHAR(<i>n</i>)
UTF-8 character string (<i>n</i> bytes long)	CHAR(<i>n</i>)
Packed decimal (<i>n</i> decimal digits)	FIXED DEC(<i>n</i> ,0)

PL/I requires that the data type, precision, length, and alignment attributes of a variable passed in a CALL statement match those of the corresponding argument on the ENTRY statement for the called procedure. If the attributes do not match, the PL/I compiler substitutes a dummy variable for the one specified in the CALL statement.

The translator generates ENTRY statements when it translates your CICS commands to PL/I CALL statements and, if there is a mismatch between the ENTRY statement specification for an argument and the variable you specify, CICS gets a dummy variable instead of yours. Although the compiler issues a warning

message when it makes such a substitution, it is easy to miss the message, and the execution results are almost never what was intended. This occurs even if there is no difference in the way the compiler implements a particular attribute value.

The ENTRY statements that the translator generates specify data type, precision, and length, using the values shown in the table above. Therefore, to prevent the compiler from substituting dummy variables, you must specify these attributes explicitly for variables used in CICS commands unless they happen to match the defaults. (Defaults come from a DEFAULT statement if you have used one, and from the compiler defaults otherwise.)

In contrast, the generated ENTRY statements do *not* specify the alignment attribute, and therefore the defaults apply. This means that alignment agreement between an argument in a CICS option and the ENTRY statement occurs only if the argument has default alignment, and happens automatically if you do not override the PL/I defaults.

Defaults at an installation can change and, therefore, the safest policy is to specify data type, length, and precision explicitly for variables used in CICS commands, and to omit the alignment specification.

If you use variable-length character strings, you need to be aware of another aspect of PL/I. PL/I prefixes character strings defined as VARYING with a two-byte length field. If you name such a string as a data-value, the data CICS receives starts with this length prefix - usually an unintended result. (The length sent to CICS is whatever you specify in the associated length option or, if you omit it, the maximum length for the string plus two for the length prefix.) Similarly, if you name the string as a *data-area*, CICS stores the information requested starting at the length prefix. CICS does not prefix character data with a length, and so this result also is usually unintended.

Assembly language argument values

In assembly language, an argument calling for a data-area, data-value, or CVDA can be any relocatable expression that refers to data of the correct type, including register forms such as 20(0,11), and forms that use the macro-replacement facilities.

You can use literal constants, such as =F'1' or =AL2(100), for data-values and sender CVDAs, but you should not use them, or any other storage that is not to be modified, for receiver arguments.

Pointer arguments, in contrast, are conveyed through a general register in CICS assembler programs and, therefore, they must be absolute expressions. For a pointer-value, you specify the number of the register that contains the address of the data (loading the register first if it does not already point to it). For a pointer-reference, you specify the register in which CICS is to return the address of the data. For example, after execution of:

```
EXEC CICS INQUIRE TASK LIST
          LISTSIZE(LISTLEN)
          SET (9)
```

the address of the task list is in register 9.

Argument lengths

Arguments in character form can be variable in length; the USERDATA option in the ACQUIRE TERMINAL command is an example. Where this occurs, CICS provides an option with which you can specify the length of the data and you must do so if you are coding in C/370.

In COBOL, PL/I, and assembler, however, you do not ordinarily need to specify this option because, if you omit it, the translator generates the length option and supplies the correct value using the language facilities. In COBOL, for example, if you write:

```
EXEC CICS ACQUIRE TERMINAL('ABCD')
         USERDATA(LOGONMSG) END-EXEC
```

the translator adds the USERDATALEN option, as if you had written:

```
EXEC CICS ACQUIRE TERMINAL('ABCD')
         USERDATALEN(LENGTH OF LOGONMSG)
         USERDATA(LOGONMSG) END-EXEC
```

Note that the translator gets the length directly from the variable name, so you must use a name with the correct length associated if you omit the length option.

In COBOL, PL/I, and assembler language, if the translator option NOLENGTH is used, the translator does not default the length options.

Null values

CICS defines a null value for most types of data. CICS sets receiver option values to the null value corresponding to the data type for the option if the option does not apply in a particular situation, and you can use them in sender options to indicate that you want no change to an option value.

The null value for each data type is listed below:

Data type	Null value
Character string (n characters long)	n blanks (X'40')
UTF-8 character string (n characters long)	n blanks (X'20')
Halfword binary	-1 (X'FFFF')
Fullword binary	-1 (X'FFFFFFFF')
Doubleword binary	-1 (X'FFFFFFFFFFFFFFFF')
Pointer (address)	X'FF000000'
CVDA (in receiver option)	DFHVALUE(NOTAPPLIC) is 1
CVDA (in sender option)	DFHVALUE(IGNORE) is 1

See the "Inquiry commands" on page 19 and "SET commands" on page 23 for more about these uses.

Exception conditions

CICS does not distinguish between SPI and API commands in the flow of control after it processes a command.

Read the material on this subject in the *CICS Application Programming Guide* if you are not familiar with it, because the information that follows is only a summary.

A program that issues a CICS command regains control at the point following the command if any of the following are true:

- The command executes normally
- You specify NOHANDLE or RESP in the command (you can specify these options in any command)
- An exception occurs for which an **IGNORE CONDITION** command has been issued

If an exception occurs for which a **HANDLE CONDITION** command is active, control goes to the point specified in the **HANDLE** command. Otherwise, CICS takes its default action for the exception. Except where specifically noted, this action is an abend. The abend codes associated with each exception condition are listed in Appendix B, "EXEC interface block (EIB) response and function codes," on page 825.

RESP and RESP2 options

CICS sets a primary and sometimes a secondary response code when it completes a command, and provides options for you to inspect them.

The primary code returned by the RESP option is the general result - either **NORMAL**, meaning that the command executed successfully, or the name of an exception condition such as **NOTAUTH** (not authorized) or **INVREQ** (invalid request). The secondary code, returned by RESP2, provides a finer level of detail.

RESP values are like CVDAs in that there is a limited set of values, each of which is named, and CICS translates the value name to its numeric equivalent. Appendix B, "EXEC interface block (EIB) response and function codes," on page 825 lists the correspondence, but use the value names in your code to keep it version- and platform-independent.

For example, here is code from a program that initializes for an application. It opens and enables a file, and then checks to ensure that the operation was successful before continuing:

```
EXEC CICS SET FILE ('TAXFILE ') OPEN ENABLED
      RESP(RC) END-EXEC.
IF RC = DFHRESP(NORMAL) PERFORM MAIN-RTN
ELSE IF RC = DFHRESP(NOTAUTH)
      PERFORM SECURITY-WARNING
      ELSE PERFORM ERR-RTN.
```

Many exception conditions can have multiple causes. If you must know the exact cause, use the RESP2 option, which you can specify whenever you have specified RESP. For example, if you want to distinguish a failure because the file was remote from other failures in the example above, you can add the RESP2 option to the SET FILE statement:

```
EXEC CICS SET FILE ('TAXFILE ') OPEN ENABLED
      RESP(RC) RESP2(RC2) END-EXEC
```

and then test explicitly for a remote file:

```
IF RC2 = 1 . . .
```

RESP2 values are numeric and predefined by CICS, like RESP values, but they are not named; you use the numeric values, as shown in the example. They are unique for a specific command, and the RESP2 value implies the RESP value, so that you

do not have to test both. They are not unique across commands, however, as RESP values are. Both are fullword binary values, defined in the same way as a CVDA in the same language:

COBOL	PIC S9(8) COMP
C and C++	long int
PL/I	FIXED BIN(31)
Assembler	F

Security checking

CICS uses an external security manager, such as RACF, to perform security checking.

Five types of security checks govern whether a particular SPI command is executed:

- Transaction
- Command
- Surrogate
- Resource
- Authtype (DB2 objects only)

CICS performs these checks only if the **SEC** system initialization parameter has a value of YES.

The **transaction** check occurs first, at task attach time, when CICS ensures that the user initiating the task has authority to use the transaction that is to be executed. This check is governed by the **XTRAN** system initialization parameter as well as **SEC**; it is skipped if the **XTRAN** value is NO. The remaining checks occur as necessary when commands are issued.

Command checking verifies that the user is authorized to use SPI commands. It is governed by the XCMD and CMDSEC system initialization parameters, and the CMDSEC option in the definition of the TRANSACTION being executed, and occurs if the XCMD value is not NO and either the CMDSEC option in the TRANSACTION is YES or the **CMDSEC** system initialization parameter is ALWAYS. If the user is not authorized, CICS rejects the command with a RESP value of NOTAUTH and a RESP2 value of 100.

If the command associates a user ID with a resource, a **surrogate** check may follow the command check. This check ensures that the user ID of the task issuing the command has authority to act as a surrogate for the user ID named in the command. It occurs only if the **XUSER** system initialization parameter is YES, and applies only to these command-option combinations:

- SET TDQUEUE with ATIUSERID
- SET DB2CONN with AUTHID or COMAUTHID
- SET DB2ENTRY with AUTHID
- SET DB2TRAN that references a DB2ENTRY containing AUTHID
- CREATE CONNECTION with SECURITYNAME
- CREATE DB2CONN with AUTHID or COMAUTHID
- CREATE DB2ENTRY with AUTHID
- CREATE DB2TRAN that references a DB2ENTRY containing AUTHID
- CREATE SESSIONS with USERID
- CREATE TDQUEUE with USERID
- CREATE TERMINAL with USERID
- CREATE BUNDLE installing an EPADAPTER that contains a USERID

CICS returns a RESP2 value of 102 if the check fails. (Additional checks on the assigned user occur on SET TDQUEUE ATUSERID, as detailed in the description of that command.)

The **resource** check verifies that the user ID has authority to use the resource in the way required by the command. Resource checking is controlled by the RESSEC option in the TRANSACTION being executed, the **RESSEC** system initialization parameter, and a system initialization parameter specific to the resource type:

- **XDCT** for transient data queues
- **XFCT** for files
- **XJCT** for journals
- **XPCT** for transactions
- **XPPT** for programs, map sets, partition sets, and exits
- **XRES** for the following CICS resources:
 ATOMSERVICE
 BUNDLE
 BUNDLEPART
 DOCTEMPLATE
 EPADAPTER
 EVENTBINDING
 JVMSERVER,
 XMLTRANSFORM
- **XTST** for temporary storage queues
- **XDB2** for DB2 entries and transactions

See Security using the XRES resource security parameter for more information about XRES.

Resource checking occurs only if the applicable resource-type system initialization system initialization parameter has a value other than NO and either the RESSEC option in the TRANSACTION is YES or the **RESSEC** system initialization parameter is ALWAYS. For commands other than **INQUIRE NEXT**, CICS rejects the command with the NOTAUTH condition and a RESP2 value of 101 if a resource check fails. During a browse, however, CICS skips resources that would fail the resource check on an ordinary INQUIRE (see “Rules for browsing” on page 22 for details).

The resources that can be protected in this way, and the SPI commands that require access authority, are shown in the table that follows. The letter in parentheses after the command indicates whether the user needs read (R), update (U), or alter (A) authority to the resource.

Resource (system initialization parameter)	Commands
Exits (XPPT option)	DISABLE PROGRAM (U) ENABLE PROGRAM (U) EXTRACT EXIT (R) INQUIRE EXITPROGRAM (R)
Files (XFCT option)	COLLECT STATISTICS FILE (R) CREATE FILE (A) DISCARD FILE (A) INQUIRE FILE (R) SET FILE (U)

Resource (system initialization parameter)	Commands
Journals (XJCT option)	COLLECT STATISTICS JOURNALNAME (R) COLLECT STATISTICS JOURNALNUM (R) DISCARD JOURNALNAME (A) INQUIRE JOURNALNAME (R) SET JOURNALNAME (U)
Programs Map sets Partition sets (XPPT option)	COLLECT STATISTICS PROGRAM (R) CREATE MAPSET (A) CREATE PARTITIONSET (A) CREATE PROGRAM (A) DISCARD PROGRAM (A) INQUIRE PROGRAM (R) SET PROGRAM (U)
Temporary storage queues (XTST option)	INQUIRE TSQUEUE (R) INQUIRE TSQNAME (R)
Transactions (XPCT option)	COLLECT STATISTICS TRANSACTION (R) CREATE TRANSACTION (A) DISCARD TRANSACTION (A) INQUIRE TRANSACTION (R) INQUIRE REQID TRANSID (R) SET TRANSACTION (U)
Transaction classes (XPCT option)	COLLECT STATISTICS TCLASS (R) COLLECT STATISTICS TRANCLASS (R) CREATE TRANCLASS (A) DISCARD TRANCLASS (A) INQUIRE TCLASS (R) INQUIRE TRANCLASS (R) SET TCLASS (U) SET TRANCLASS (U)
Transient data queues (XDCT option)	COLLECT STATISTICS TDQUEUE (R) CREATE TDQUEUE (A) DISCARD TDQUEUE (A) INQUIRE TDQUEUE (R) SET TDQUEUE (U)
DB2ENTRYs (XDB2 option)	CREATE DB2ENTRY (A) CREATE DB2TRAN (A) INQUIRE DB2ENTRY (R) INQUIRE DB2TRAN (R) SET DB2ENTRY (U) SET DB2TRAN (U)
DB2TRANs (XDB2 option)	CREATE DB2ENTRY (A) CREATE DB2TRAN (A) INQUIRE DB2ENTRY (R) INQUIRE DB2TRAN (R) SET DB2ENTRY (U) SET DB2TRAN (U)

Resource (system initialization parameter)	Commands
CICS resources that are subject to XRES security checks (XRES option)	CREATE ATOMSERVICE (A) CREATE BUNDLE (A) CREATE DOCTEMPLATE (A) CREATE JVMSERVER (A) DISCARD ATOMSERVICE (A) DISCARD BUNDLE (A) DISCARD DOCTEMPLATE (A) DISCARD JVMSERVER (A) DOCUMENT CREATE (R) DOCUMENT INSERT (R) INQUIRE ATOMSERVICE (R) INQUIRE BUNDLE (R) INQUIRE BUNDLEPART (R) INQUIRE CAPTURESPEC (R) INQUIRE DOCTEMPLATE (R) INQUIRE EPADAPTER (R) INQUIRE EVENTBINDING (R) INQUIRE EVENTPROCESS (R) INQUIRE JVMSERVER (R) INQUIRE MQINI (R) INQUIRE XMLTRANSFORM (R) SET ATOMSERVICE (U) SET BUNDLE (U) SET DOCTEMPLATE (U) SET EPADAPTER (U) SET EVENTBINDING (U) SET EVENTPROCESS (U) SET JVMSERVER (U) SET XMLTRANSFORM (U)

Authtype checking applies to DB2CONN_s, DB2ENTRY_s, and DB2TRAN_s only. For more information, see DB2 overview in the DB2 Guide.

The QUERY SECURITY command

You can find out whether you are authorized to access a resource or to issue a system programming command by issuing the QUERY SECURITY command. This is not an SPI command and does not access any resources, and so never raises a NOTAUTH condition. It is described in the *CICS Application Programming Reference*.

Inquiry commands

You can use the system programming commands to inquire about the definition and status of most resources that are defined to CICS, and about many elements of the CICS system as well.

You cannot inquire on the following CICS resources:

- LSRPOOL
- MAPSET
- PARTITIONSET
- TYPETERM

For most resource types, the options in the **INQUIRE** command correspond to specific elements in the definition of that resource. Such options usually have the

same or similar names in the **INQUIRE** command and in the resource definition. Where they do not, the option text for the command notes the corresponding definition attribute. Often, for more information about the meaning of an option value, see the definition of the resource in the *CICS Resource Definition Guide*.

In addition to CICS resources, you can inquire on elements of the CICS system such as the CICS dispatcher. Most system elements that you can inquire about correspond to system initialization parameters. For more information about them, see the system initialization parameter descriptions in *CICS System Definition Guide*.

Certain considerations apply to all the inquiry commands, which are principally the **INQUIRE** commands, but also include **COLLECT STATISTICS**, **EXTRACT STATISTICS**, and **EXTRACT EXIT**.

- **Exception conditions:** CICS returns no information when an exception condition occurs; data-areas named in receiver options are unchanged.
- **Exclusive control:** A task that inquires about a resource, system setting, or system component does not get exclusive control of the object of the inquiry. Another task or system event might change the information returned at any time. The resource currently being inquired on must not be deleted because the current resource is used to position to the next resource on a subsequent **GETNEXT** command. The resource can be deleted only after the subsequent **GETNEXT** command, because it is no longer required for positioning within this browse request.
- **Browsing:** Resources that support browsing can be retrieved sequentially, as explained in “Browsing resource definitions.”
- **Inapplicable options:** If you specify a receiver option that does not apply to the resource about which you are inquiring, CICS generally returns the appropriate “null value”, as defined in “Null values” on page 14. (In a few cases, an exception is raised; these cases are noted in the command descriptions.)
For example, if you include **BLOCKFORMAT** in an **INQUIRE TDQUEUE** command that specifies an intrapartition transient data queue, CICS returns the value **NOTAPPLIC** to the CVDA you provide, because **BLOCKFORMAT** is valid only for extrapartition queues.

Browsing resource definitions

The **INQUIRE** commands that apply to resources ordinarily retrieve information about a single resource that you name when you issue the command. However, there is another form that enables you to browse through some or all of the definitions of a given type.

The documentation for each **INQUIRE** command states whether or not the browse options are supported for that resource type.

There are three steps to browsing resource definitions:

1. Starting a browse of the resource definitions.
2. Retrieving the next resource.
3. Ending the browse of the resource definitions.

Starting a browse

You issue the **INQUIRE** command with an additional option, **START**, to set up the browse. This command does not produce any information; it just tells CICS what you are going to do.

The general form of the command is:

```
INQUIRE resource-type START
```

In addition to the START option, there are several differences in the way you issue this setup command from the normal syntax:

- You identify the resource type only, without providing a resource name; that is, the resource type appears without its customary data-value.
- You omit all of the options in which CICS returns information to you.
- You also omit options that send information to CICS, other than the resource type. (INQUIRE EXITPROGRAM and INQUIRE UOWENQ are exceptions to this rule; you can limit the browse by supplying additional information on the START, as explained in the descriptions of these commands.)

Generally, CICS returns resource definitions to you in the order it keeps them internally. You cannot control this order, and you should not depend on it always being the same. For a few resource types, however, CICS returns definitions in alphabetic order of resource name. These are:

- DB2ENTRYs and DB2TRANs
- Programs, map sets, and partition sets
- Temporary storage queues
- Transactions
- Transaction classes

For these resources only, you can specify a starting point for the browse with the AT option on the INQUIRE START:

```
INQUIRE resource-type START AT(data-value)
```

The AT data-value is the name at which you want to start. It must be in the correct format for a name of the resource type being browsed, but it does not have to correspond to an installed resource; it is used only to start the browse at the proper point in the resource list. CICS restricts the definitions that it returns on your INQUIRE NEXT commands to resources with names equal to or greater (in the collating sequence) than the value you provide.

JVM profiles are also returned in alphabetic order of resource name, but you cannot use the AT option with the INQUIRE JVMPROFILE START command.

Retrieving the next resource

In the second step of a browse, you issue the INQUIRE command repetitively with another new option, NEXT. CICS returns one resource definition for each INQUIRE NEXT.

The general format is:

```
INQUIRE resource-type(data-area) NEXT option...option
```

Apart from the addition of NEXT, the options are almost the same on an INQUIRE NEXT as on a single INQUIRE for the same type of resource. Again, however, there are some differences:

- Instead of specifying the name of the resource (a data-value), you provide a **data-area** of the same length for CICS to return the name of the next resource to you.

- Options by which CICS returns data to you are used in the same way as on the single-resource form.
- A few options, such as the CONNECTION option on **INQUIRE MODENAME**, change their roles in a browse. These differences also are noted in the commands to which they apply.

You repeat the **INQUIRE NEXT** command until you have seen the resource definitions you want or have exhausted the definitions. After you have retrieved the last of them, CICS raises the END condition on subsequent INQUIRE NEXTs, leaving any data-areas you provided unchanged. However, you do not have to retrieve all the definitions; you can stop the browse at any time.

Ending the browse

Stopping the browse is the final step.

To do so you issue an INQUIRE for the resource type with just the END option, as follows:

```
INQUIRE resource-type END
```

Browse example

Here is an example of a typical browse sequence. This code retrieves the names of all the files installed in the system and calls a subroutine to process information about the recovery characteristics if the file is open.

```
EXEC CICS INQUIRE FILE START END-EXEC.
PERFORM UNTIL RESP CODE = DFHRESP(END)
  EXEC CICS INQUIRE FILE(FILENAME) NEXT
    OPENSTATUS(OPENSTAT)
    RECOVSTAT(RCVRSTAT)
    FWDRECSTATUS(FWDSTAT)
    RESP(RESPCODE) END-EXEC
  IF RESP CODE = DFHRESP(NORMAL)
    IF OPENSTAT = DFHVALUE(OPEN)
      CALL RCVY-RTN USING RCVRSTAT FWDSTAT
    END-IF
  ELSE CALL ERROR-RTN END-IF
END-PERFORM.
EXEC CICS INQUIRE FILE END END-EXEC.
```

Rules for browsing

There are some rules you should know about browsing resource definitions.

1. Your position in a browse is associated with your task, so that it is preserved across LINK and XCTL commands. Programs that run as part of a program list table (PLT) during CICS initialization or termination run under a single task. Consequently, they should terminate explicitly any browse they begin, in order not to conflict with other programs in the same PLT.
2. A task can browse more than one type of resource at the same time, but can have only one browse in progress for a particular resource type.
3. A SYNCPOINT command does not end a browse or affect your position in it.
4. Resource definitions are not locked during a browse, and another task may change the definitions while you are inquiring on them.
5. Nonetheless, you should always end a resource browse explicitly, rather than allowing end-of-task processing to do so implicitly, because a browse holds control blocks that other tasks may require for browsing.
6. INQUIRE NEXT commands usually do not cause a task switch. Therefore, a task browsing a long list of resources may exceed the runaway task interval

without giving up control, causing CICS to abend it with an AICA code. If this occurs, you need to intersperse a SUSPEND command periodically among your INQUIRE NEXTs.

7. During a browse in a task for which resource security checking is in effect, CICS returns only those definitions that the user is authorized to see. The others are skipped without any indication.

Exception conditions for browsing

Two conditions can occur on the browse forms of an INQUIRE command, in addition to those that apply to the single-resource form of the command.

These conditions are as follows:

END

RESP2 values:

- 2 INQUIRE NEXT has been issued, but there are no more resource definitions of the type being browsed.

ILLOGIC

RESP2 values:

- 1 A START has been given when a browse of the same resource type is already in progress, or a NEXT or an END has been given without a preceding START.

SET commands

You can change most of the system elements and resource definitions about which you can inquire, although in general you cannot change as many option values as you can retrieve. Changes are made with a SET command naming the resource or system element.

Like the INQUIRE commands, SET commands follow some general rules:

- **Exceptions:** When a SET command results in an exception condition, CICS makes as few of the requested changes as possible. To establish which, if any, changes have been made, you can issue the corresponding INQUIRE command.
- **Permanence:** If you change a system setting or resource definition element that is ordinarily recorded in the CICS global catalog, the change is also recorded in the catalog and thus preserved over a warm or emergency restart. If the information is not ordinarily recorded, it lasts only for the current execution of CICS. In a cold or initial start, the catalog information is discarded and all effects of earlier SET commands are lost.
- **Recoverability:** SET commands are not recoverable. Their effects are not backed out if the task that issued them abends or issues a SYNCPOINT ROLLBACK command. Consequently, SET commands do not lock resources, and you do not need to precede a SET with the corresponding INQUIRE command.
- **“No change” values:** Except where there is a default value for an option, CICS does not change the value associated with an option that you omit. However, there is a second way to indicate that you want no change. If you specify the null value in a sender option that is not required, CICS leaves the option value unchanged. Although you can get the same effect by omitting the option if there is no default, the ability to specify a “no change” value allows you to vary the options in a command as well as the option values, simplifying your code in some situations.

For example, suppose you needed to change many different combinations of options, depending on the outcome of some calculations. Your code might look something like this:

```
IF ... MOVE DFHVALUE(NOTDELETABLE) TO DEL
ELSE MOVE DFHVALUE(IGNORE) TO DEL.
IF ... MOVE 2 TO POOL
ELSE MOVE -1 TO POOL.
IF ... MOVE 'TAXID.MASTER' to DSN
ELSE MOVE SPACES TO DSN.
EXEC CICS SET FILE('TAXMSTR ') DELETE(DEL)
        LSRPOOLNUM(POOL) DSNAME(DSN) END-EXEC.
```

See “Null values” on page 14 for more about null values.

Note: There are a few options, such as the NEXTTRANSID option in a SET TERMINAL command, for which blanks (the null value for a character field) are a meaningful value. For these options, there is no null value, and you must omit the option if you do not want to change its value; these cases are noted in the option descriptions.

Creating resource definitions

CREATE commands allow you to add resource definitions to the local CICS region by using a program, so that you can write applications to administer a running CICS system. These definitions are equivalent to those produced by CEDA transactions. They are recorded in the CICS global catalog and persist over a warm or emergency restart.

However, CREATE commands neither refer to nor record in the CICS system definition (CSD) file. Consequently, the resulting definitions are lost on a cold or initial start, and you cannot refer to them in a CEDA transaction.

You can create definitions for the following types of resources:

- ATOMSERVICES
- Bundles
- Connections
- CorbaServer (CORBASERVER)
- DB2 connection
- DB2 resources (DB2ENTRYs DB2TRANs)
- Deployed JAR files (DJAR)
- Document templates
- ENQ models
- Files
- IPIC connections
- Journal models
- JVM servers
- LSR pools
- LIBRARY concatenations
- Map sets
- Partition sets
- Partners
- PIPELINEs
- Process types
- Profiles
- Programs
- Request models
- Sessions
- TCP/IP service

- Temporary storage queue models
- Transient data queues
- Terminals
- Terminal types (TYPETERMs)
- Transaction classes
- Transactions
- URIMAPs
- WEBSERVICES
- WebSphere MQ connections (MQCONN)

A CREATE command corresponds to a combined CEDA DEFINE and INSTALL, except for not updating the CSD file. If there is no resource of the same name and type already installed, the new definition is added to the resources of your CICS region. (Definitions always apply to the local CICS region, even if they describe resources located on a remote system.) If the resource was already installed, the new definition replaces the old one, and an implicit discard of the old resource occurs as well. In this case, most restrictions that would apply to a DISCARD command naming the same resource apply to the CREATE.

During the processing, CICS sync points your task, as if a SYNCPOINT command had been issued along with the CREATE. Changes made to recoverable resources between the CREATE and task start (or the most recent syncpoint) are committed if processing is successful, and rolled back if not. (For TERMINAL definitions and CONNECTION-SESSIONS definitions that require more than one CREATE command to complete, the syncpoint takes place on the final CREATE of the sequence.)

If an error is detected before installation processing begins, installation is not attempted. CICS raises an exception condition and returns control to the issuing task without sync pointing. However, some errors are detected later in the process and cause rollback, and all successful CREATEs cause a commit. Tasks using these commands need to be written with these commit effects in mind.

In addition, the implied syncpoint means that CREATE commands cannot be issued in a program invoked by a distributed program link unless the LINK command specifies SYNCONRETURN, in a program with an EXECUTIONSET value of DPLSUBSET, or in any other situation where syncpoint is not allowed.

CREATE commands can be executed at any time after the start of the third phase of CICS initialization. This means they can be used in programs specified in the second section of the program load table for postinitialization (PLTPI) as well as during normal CICS execution.

The ATTRIBUTES option

The specifics of the resource definition that a CREATE or a CSD command installs are conveyed through the ATTRIBUTES option value, which is a character string listing the attributes of the resource.

You specify attributes and attribute values in text form, in the same way that you do on a CEDA DEFINE screen. This character string is analyzed at the time the CREATE or CSD command is executed, and consequently must consist entirely of text, rather than variable names, in a single string. The syntax in the string is provided for each CREATE or CSD command, using the same conventions as

command syntax, except for the attribute values as noted below. However, the contents are *not* parsed by the translator, which checks only the command syntax, shown in the main diagram.

Attribute values appear essentially as they do on CEDA DEFINE screens. However, because DEFINE screens are preformatted and ATTRIBUTES strings are not, you need to know the following rules:

- Attributes may appear in any order (you do not have to follow the order in the syntax diagram or in the CEDA command).
- The name of an attribute must be that shown in the syntax diagram or the abbreviation permitted in the corresponding CEDA DEFINE entry (see the discussion of DEFINE in Resource management transaction CEDA commands in the Resource Definition Guide).

Note: Abbreviations can change from release to release, and thus full spellings are safest.

- The attribute string is not converted to uppercase, in contrast to inputs to CEDA and the DFHCSDUP utility. Attribute names are recognized regardless whether you use upper, lower, or mixed case, as are value names assigned by CICS (those shown in uppercase letters in the syntax diagram). However, other character values—resource names and message text, for example—are taken as is, so that you need to supply them in the intended case.
- The argument value, if any, must follow the rules for the same attribute in a CEDA DEFINE panel. Where there are a limited number of possible values, they are listed in the attributes diagram in uppercase. Otherwise the diagram indicates only the form of the value, using the following conventions:

char*n* A character string of length *n* or, where the argument can be of variable length, of maximum length *n*.

hex*n* A string of hexadecimal characters of length *n* or, where the argument can be of variable length, of maximum length *n*.

n1-n2 A number in the range *n1* to *n2*.

Note: You can omit trailing blanks in character arguments, trailing X'00's in hexadecimal arguments, and leading zeros in numeric arguments.

- You can use one or more blanks to separate attributes for readability, but a blank is required only between an attribute that has no argument and the next attribute. Commas and other separators are not allowed. Blanks may also appear between an attribute name and the parentheses that surround its argument, and between the parentheses and the argument value, but they are not necessary. Thus both of these, and similar combinations, are correct:

```
ATTRIBUTES ( 'UCTRAN (NO)RTIMEOUT (10 )' )  
ATTRIBUTES(' UCTRAN(NO) RTIMEOUT( 10) ' )
```

- No quote marks are required within the attribute string (you need them around the whole string if you use a literal, as in the example above). If you want quotes within your text—in the DESCRIPTION attribute, for example—use two quote characters for each one that you want to appear in the text, as you do in literal constants that contain quotes.
- Very few attributes require specification, and omitting one is equivalent to not keying a value for it on a CEDA screen. Where the default value is always the same, it is shown in the diagram in the same way as in syntax diagrams. However, some defaults depend on the values of other attributes, and these are not shown. (You cannot define your own defaults for CREATE commands because they do not use the CSD file.)

- For some resource types, you can use defaults for all attributes. If you want to do this, set the length of the string to zero in the ATTRLEN option. You must still specify the ATTRIBUTES option in this case, even though the data-value you provide is not examined.
- You can omit the ATTRLEN option when it is not zero if it is the length of the variable specified in ATTRIBUTES and you are not coding in C/370, as explained in “Argument lengths” on page 14.

If you make an error in the ATTRIBUTES string, CICS raises the INVREQ condition with an appropriate RESP2 value. Appendix C, “RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands,” on page 837 lists the RESP2 values that apply.

Discarding resource definitions

The DISCARD command deletes the definition of a resource installed in the local CICS system, so that the system no longer has access to the resource, or makes a model ineligible for use as a model.

It reverses the effect of the installation of the resource, which can occur at system startup, through a subsequent CREATE command or CEDDA transaction, by an automatic installation process, or by a CICSplex SM BAS command.

Each DISCARD command removes the definition of one resource. You can remove definitions for the following types of resources:

- Atom feeds
- Autoinstall models for terminals
- Bundles
- Connections
- CorbaServer (CORBASERVER)
- DB2Conns
- DB2EntryS
- DB2Trans
- Deployed JAR files (DJAR)
- Document templates
- ENQ models
- EP adapters
- Event bindings
- Files
- IPIC connections
- Journals and journal models
- LIBRARY concatenations
- WebSphere MQ connections
- Partners
- PIPELINEs
- Process types
- Profiles
- Programs, map sets, and partition sets
- Request models
- TCP/IP service
- Temporary storage queue models
- Terminals
- Transaction classes
- Transactions
- Transient data queues
- URIMAPs

- WEBSERVICEs

You cannot discard a resource that is currently in use. For example, you cannot discard a PROFILE definition if some installed TRANSACTION definition still points to it, or a FILE that is open, or a TRANSACTION that is scheduled for execution.

In addition, some resources are not eligible for discard at all. These include resources whose names begin with the letters DFH (reserved for CICS-supplied definitions), and transactions whose names begin with C (also reserved for CICS).

Some DISCARD commands cause a syncpoint on behalf of the issuing task, as the CREATE commands do. For these commands, the discussion of syncpoint considerations on page Creating resource definitions, applies.

DISCARD commands are recorded in the CICS catalog, so that their effects persist over a warm or emergency restart, but they do not modify the CSD file and are therefore lost on a cold or initial start.

Exit-related commands

In CICS, an exit is installation-supplied code that is invoked either at specific, CICS-defined points within CICS system code, or by an application request that uses the exit mechanism, such as a DB2 or IMS request.

There are two types: global user exits and task-related user exits. Global user exits are always invoked at CICS-defined points in system code; task-related exits can be invoked both ways. The *CICS Customization Guide* lists the points in CICS code at which global exits may be invoked, describes how and when task-related exits are driven, and gives full details for programming exits.

Five SPI commands are related to exits:

- ENABLE PROGRAM
- DISABLE PROGRAM
- EXTRACT EXIT
- RESYNC ENTRYNAME
- INQUIRE EXITPROGRAM

You can use them in any language supported by CICS, even though the exit itself must be coded in assembler.

Defining exits

The only way to define an exit in CICS - that is, to install it so that the code gets executed - is to issue the **ENABLE PROGRAM** command.

Similarly, the only way to delete the definition is to issue the corresponding **DISABLE PROGRAM EXITALL** command or shut down the system. Exit definitions last only for the current execution of CICS. They are not recorded in keypoints, the CICS global catalog, or the CSD file, and therefore do not survive a shutdown of any kind.

ENABLE and **DISABLE PROGRAM** commands affect only the CICS region in which they are issued. Even if CICS system code or exit program code is shared among several executing CICS regions, the exit must be defined and deleted separately in each region that uses it.

Moreover, these commands are not recoverable; their effects are not backed out if the task that issued them fails or issues a **SYNCPOINT ROLLBACK** command.

Exit names

The code that an exit executes is contained in one or more ordinary load modules; a module can be used both by an exit and a user transaction.

You identify the first module to be executed in an exit by naming it in the **PROGRAM** option of the **ENABLE PROGRAM** command that creates the exit. The exit can execute other modules as well, but you tell CICS where to start, just as you name only the first program to be executed in a **TRANSACTION** definition.

Exits are named by the **ENTRYNAME** value in the initial **ENABLE PROGRAM** command, not the **PROGRAM** value, although you can omit the **ENTRYNAME** option and allow its value to default to the **PROGRAM** value. Exit names must be unique, however, and if a program is used first by more than one exit, only one of them can be named by default in this way. Moreover, even when an exit and its first program have the same name, they are separate entities of different types.

Because of this default (and some history), CICS requires that you always identify an exit in the same way that you did in the **ENABLE PROGRAM** command that created it—that is, by coding (or omitting) the same **PROGRAM** and **ENTRYNAME** values. **RESYNC ENTRYNAME** is an exception; you specify the exit name in the **ENTRYNAME** option, regardless of whether you used **ENTRYNAME** or **PROGRAM** to assign the name initially. Also, in the **INQUIRE EXITPROGRAM** command, the option that names the initial program is **EXITPROGRAM** rather than **PROGRAM**.

Like modules invoked by user transactions, load modules used by exits must be defined as **PROGRAM** resources, either explicitly or by autoinstallation, and they must have an **ENABLESTATUS** value of **ENABLED** at the time of invocation. In addition, the initial program for an exit must be in **ENABLED** status at the time of the **ENABLE PROGRAM** command that creates the exit. However, the **ENABLESTATUS** of a program is independent of any exits that use it, and it is not affected by **ENABLE** and **DISABLE PROGRAM** commands that refer to it.

CICS threadsafe commands in the SPI

If you write and define a CICS program as threadsafe, it can receive control on an open transaction environment (OTE) TCB.

To obtain the maximum performance benefit from OTE, write your CICS programs in a threadsafe manner to avoid CICS having to switch TCBs. However, be aware that not all **EXEC CICS** commands are threadsafe, and issuing any of the non-threadsafe commands causes CICS to switch your task back to the **QR TCB** to ensure serialization. The commands that are threadsafe are indicated in the command syntax diagrams in this programming reference with the statement: "This command is threadsafe".

For a list of the system programming interface (SPI) commands that are threadsafe, see Appendix D, "Threadsafe SPI commands," on page 863.

For similar statements about the application programming interface (API), see the *CICS Application Programming Reference*.

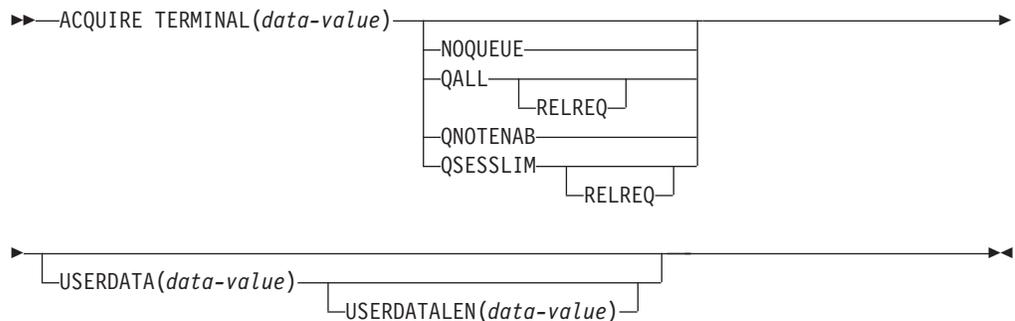
Chapter 2. System commands

Alphabetic listing of CICS system commands.

ACQUIRE TERMINAL

Acquire a session with a terminal.

ACQUIRE TERMINAL



Conditions: INVREQ, LENGERR, NOTAUTH, TERMIDERR

Description

The ACQUIRE TERMINAL command enables you to tell CICS to acquire a session with a particular terminal.

The terminal you specify must be a z/OS Communications Server terminal, and it cannot be an APPC, LU6.1, or IRC session. It must already be defined to CICS, either in an installed TERMINAL definition or by the autoinstall process, and it must be local to the system on which the ACQUIRE TERMINAL is issued, not remote.

This means that, if the terminal was autoinstalled, you must issue the ACQUIRE command before CICS deletes the terminal definition.

CICS normally deletes an autoinstalled terminal definition if the session ends and is not reestablished within the interval specified in the AIRDELAY value in the system initialization table. The terminal does not have to be reacquired within this interval, however; after you issue the command, CICS suspends its time-out and does not delete the definition while waiting for the session to be reestablished.

CICS processes an ACQUIRE command by sending a SIMLOGON request to z/OS Communications Server (the queueing options on the command are for z/OS Communications Server use and correspond to those on a SIMLOGON request). The task that issued the command is dispatchable as soon as this occurs. It is not notified of the eventual result of the z/OS Communications Server request, nor when the terminal is acquired, and the terminal does not become associated with the task.

The request is sent straight to z/OS Communications Server unless the terminal is already in session with the requesting CICS system. If it is, and NOQUEUE or QNOTENAB are present, CICS rejects the request as invalid (because a SIMLOGON would fail under these circumstances). Otherwise, CICS stores the request until the terminal's current session ends and then sends it to z/OS Communications Server. For this reason, requests may be queued by z/OS Communications Server in a different order from the order in which they were originally issued.

After it has been issued, an ACQUIRE TERMINAL request cannot be canceled, and you cannot ordinarily determine whether an ACQUIRE TERMINAL has been issued for a particular terminal.

Options

NOQUEUE

specifies that z/OS Communications Server should not queue the request. Consequently, the ACQUIRE succeeds only if the terminal is immediately available.

QALL

specifies that z/OS Communications Server should queue the request if the terminal is not enabled for sessions or is at its session limit (that is, in session with another z/OS Communications Server application).

QNOTENAB

specifies that z/OS Communications Server should queue the request only if the terminal is not enabled for sessions.

QSESSLIM

specifies that z/OS Communications Server should queue the request only if the terminal is at its session limit (that is, in session with another z/OS Communications Server application).

RELREQ

is meaningful only if the QALL or QSESSLIM option is set. The RELREQ option specifies that, if the requested terminal is already in session with another z/OS Communications Server application, that application is notified of your request via its RELREQ exit routine. If RELREQ is not specified, the other application is not notified.

If the other application is a CICS system, the RELREQ value of the terminal definition in that system determines whether the request to release the terminal is honored. RELREQ is specified on the TYPETERM definition associated with the terminal.

TERMINAL (*data-value*)

is the 4-character identifier of the terminal with which CICS is to acquire a session.

USERDATA (*data-value*)

specifies the data area containing the logon user data, if any. z/OS Communications Server simulates a logon when CICS asks to acquire a terminal. This data corresponds to user data that sometimes accompanies a real logon. z/OS Communications Server passes it to the application (in this case, the requesting CICS system) when the terminal has been acquired successfully. See the description of the EXTRACT LOGONMSG command in the *CICS Application Programming Reference* for programming information.

USERDATALEN(*data-value*)

specifies the length, as a halfword binary value, of the user data. Because of a z/OS Communications Server limitation, the maximum length of the user data is restricted to 255 bytes.

Conditions**INVREQ**

RESP2 values:

- 2 The terminal is a remote terminal.
- 3 The terminal is LU6.1, APPC, IRC or a non-z/OS Communications Server device.
- 4 The terminal is not in service; that is, it is not available for use.
- 5 z/OS Communications Server is not open.
- 7 CICS is already in the process of acquiring this session.
- 8 NOQUEUE and QNOTENAB options are invalid for a logged-on device.

LENGERR

RESP2 values:

- 6 Out-of-range value supplied in the USERDATALEN option.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

TERMIDERR

RESP2 values:

- 1 The terminal cannot be found.

COLLECT STATISTICS

Retrieve the current statistics for a single resource, or global statistics for a class of resources.

COLLECT STATISTICS

►► COLLECT STATISTICS SET(*ptr-ref*) last time | resource types | ►►

Conditions: INVREQ, IOERR, NOTAUTH, NOTFND

Resource types:

AUTOINSTALL	
CONNECTION(<i>data-value</i>)	
CORBASERVER(<i>data-value</i>)	BEAN(<i>data-value</i>)
DB2CONN	
DB2ENTRY(<i>data-value</i>)	
DISPATCHER	
ENQUEUE	
FILE(<i>data-value</i>)	
JOURNALNAME(<i>data-value</i>)	
JOURNALNUM(<i>data-value</i>)	
JVMPOOL	
JVMPROFILE(<i>data-value</i>)	
JVMPROGRAM(<i>data-value</i>)	
LSRPOOL(<i>data-value</i>)	
MONITOR	(<i>data-value</i>)
MVSTCB	(<i>data-value</i>)
POOL(<i>data-value</i>)	TARGET(<i>data-value</i>)
PROGAUTO	
PROGRAM	(<i>data-value</i>)
RECOVERY	
REQUESTMODEL(<i>data-value</i>)	
STATS	
STORAGE	(<i>data-value</i>)
STREAMNAME	(<i>data-value</i>)
SUBPOOL(<i>data-value</i>)	
SYSDUMPCODE	(<i>data-value</i>)
TABLEMGR	
TARGET(<i>data-value</i>)	NODE(<i>data-value</i>)
TASKSUBPOOL	
TCLASS(<i>data-value</i>)	
TCPIP	
TCPIPSERVICE(<i>data-value</i>)	
TDQUEUE	(<i>data-value</i>)
TERMINAL(<i>data-value</i>)	
TRANCLASS(<i>data-value</i>)	
TRANDUMPCODE	(<i>data-value</i>)
TRANSACTION	(<i>data-value</i>)
TSQUEUE	
VTAM	

Description

The **COLLECT STATISTICS** command returns to the invoking application the current statistics for a particular resource, or overall statistics for the resources of a given

type. For example, you can get the statistics for global transaction activity in your CICS system (such as the total number of transactions attached), or you can specify a single transaction that you are interested in (such as CEMT).

Attention: The list of resources supported by the COLLECT STATISTICS command is now closed. All new resources introduced from CICS TS for z/OS, Version 3.1 onwards are supported by the “EXTRACT STATISTICS” on page 207 command, which operates in the same way.

The statistics that CICS gives you are those that have been accumulated after the expiry of the last statistics collection interval, end-of-day expiry, or requested reset. (Statistics already written to the SMF data set cannot be accessed.) The COLLECT STATISTICS command does not cause the statistics counters to be reset.

CICS obtains enough storage for the data returned from this command, and returns a pointer to this area. The first two bytes of the area contain its length. This storage can be reused by subsequent COLLECT STATISTICS commands, so you should store elsewhere any data that is required beyond the next issue of the command. CICS releases this storage at task termination.

Table 1 contains Product-sensitive Programming Interface information. See the Notices section at the end of this manual for an explanation of Programming Interface information.

Not all resource types provide both global and specific statistics. Table 1 tells you which statistics are available for each resource type, and gives the copybook name for each set of available statistics. The copybooks define the format of the returned statistics. Where no copybook name is given in the global statistics column, global statistics are unavailable for the resource type; similarly, where there is no entry in the specific statistics column, you cannot get statistics for an individual resource.

Table 1. Resource types and statistics

Resource type	Statistic type	Global statistics	Specific statistics
AUTOINSTALL	Terminal autoinstall	DFHA04DS	-
BEAN	Enterprise beans	-	DFHEJBDS
CONNECTION	ISC/IRC system and mode entries	-	DFHA14DS
CORBASERVER	CorbaServer entries	-	DFHEJRDS
DB2CONN	DB2 [®] Connection	DFHD2GDS	-
DB2ENTRY	DB2 Entry	-	DFHD2RDS
DISPATCHER	Dispatcher	DFHDSGDS	-
ENQUEUE	Enqueue	DFHNQGDS	-
FEPI CONNECTION	FEPI Connection	-	DFHA23DS
FEPI POOL	FEPI Pool	-	DFHA22DS
FEPI TARGET	FEPI Target	-	DFHA24DS
FILE	File control	-	DFHA17DS
JOURNALNAME	Journalname	-	DFHLGRDS
JOURNALNUM	Journalname	-	DFHLGRDS
JVMPPOOL	JVM pool	DFHSJGDS	-
JVMPROFILE	JVM profiles	-	DFHSJRDS

Table 1. Resource types and statistics (continued)

Resource type	Statistic type	Global statistics	Specific statistics
JVMPROGRAM	JVM programs	-	DFHPPGRDS
LSRPOOL	LSR pools	-	DFHA08DS
MONITOR	Monitor	DFHMNGDS	DFHMNTDS
MVSTCB	MVS™ TCB	DFHDSTDS	DFHDSRDS
PROGAUTO	Program autoinstall	DFHPPGDS	-
PROGRAM	Program	DFHLDGDS	DFHLDRDS
RECOVERY	Recovery manager	DFHRMGDS	-
REQUESTMODEL	Request models	-	DFHIIKDS
STATS	Statistics	DFHSTGDS	-
STORAGE	Storage manager (SM)	DFHSMGDS	DFHSMDDS
STREAMNAME	Log stream	DFHLGGDS	DFHLGSDS
SUBPOOL	SM domain subpool	-	DFHSMDDS
SYSDUMPCODE	Dump (system)	DFHSDGDS	DFHSDRDS
TABLEMGR	Table manager	DFHA16DS	-
TASKSUBPOOL	SM task subpool	DFHSMTDS	-
TCLASS	Transaction class	-	DFHXMCDL
TCPIP	Sockets domain	DFHSOGDS	-
TCPIPSERVICE	TCP/IP service	-	DFHSORDL
TDQUEUE	Transient data	DFHTQGDS	DFHTQRDS
TERMINAL	Terminals	-	DFHA06DS
TRANCLASS	Transaction class	-	DFHXMCDL
TRANDUMPCODE	Dump (transaction)	DFHTDGDS	DFHTDRDS
TRANSACTION	Transaction manager	DFHXMGDS	DFHXMRL
TSQUEUE	Temporary storage	DFHTSGDS	-
z/OS Communications Server	VTAM®	DFHA03DS	-

Copybooks are provided in ASSEMBLER, C, COBOL, and PL/I. The names of the copybooks are the same in each language. You can find them in the following libraries:

ASSEMBLER	CICSTS42.CICS.SDFHMAC
C	CICSTS42.CICS.SDFHC370
COBOL	CICSTS42.CICS.SDFHCOB
PL/I	CICSTS42.CICS.SDFHPL1

Note: Some of the copybooks contain packed fields. Before these fields are used, they should be checked for hexadecimal zeros. The COBOL versions of the fields have been redefined as numeric with a suffix of -R for this purpose.

For further information about these copybooks, see the *CICS Performance Guide*.

Options

AUTOINSTALL

requests global statistics on autoinstall.

BEAN(*data-value*)

requests statistics for an enterprise bean; data-value is the 1–240 character name of the bean.

Because a bean of the same name might exist in multiple CorbaServers, if you specify this option you must also specify CORBASERVER.

CONNECTION(*data-value*)

requests statistics for a connection to a remote system or region; data-value is the 4-character identifier (from its CONNECTION definition) of the system or region.

CORBASERVER(*data-value*)

requests statistics for a CorbaServer; data-value is the 4-character name of the CorbaServer.

DB2CONN

requests statistics for the CICS DB2 connection including information for pool threads and command threads.

DB2ENTRY(*data-value*)

requests statistics for a DB2ENTRY; data-value is the 8-character identifier of the DB2ENTRY (from its DB2ENTRY definition).

DISPATCHER

requests global statistics on the dispatcher domain.

ENQUEUE

requests global statistics for enqueue requests.

FILE(*data-value*)

requests statistics for a file; data-value is the 8-character identifier of the file (from its FILE definition).

JOURNALNAME(*data-value*)

requests statistics for a CICS journal; data-value is an 8-character journal name. CICS returns the address of the area of storage that contains the requested statistics.

To collect statistics for journals defined using the journal numbering convention (for example, for the auto journals defined in file resource definitions), specify the name as DFH*nn*, where nn is the journal number in the range 01 - 99.

Note: Specifying DFHJ01 returns statistics written to a user journal of that name, *not* the system log.

JOURNALNUM(*data-value*)

requests statistics for a journal; data-value is the number of the journal, in halfword binary format. Journal numbers range from 1 to 99. CICS returns the address of the area of storage that contains the requested statistics.

Specifying JOURNALNUM(1) returns statistics for journal DFHJ01. Note that this is *not* the system log.

Specifying identifiers in the range 1—99 returns statistics for journals DFHJ01—DFHJ99.

Note: JOURNALNUM continues to be supported for compatibility with earlier releases of CICS. However, the statistics returned are CICS log manager statistics, *not* journal control statistics. You can map the data at the address returned only by using the DFHLGRDS DSECT which replaces the DFHA13DS DSECT supported at earlier releases.

When you make changes to application programs that use JOURNALNUM, it is preferred to use the JOURNALNAME option.

JVMPOOL

requests statistics for the JVM pool, if one exists. (There is no identifier on JVMPOOL: a CICS region can have only one JVM pool.) CICS returns the address of the area of storage that contains the requested statistics.

JVMPROFILE (*data-value*)

requests statistics for a JVM profile. The data-value is the 8-character name of the JVM profile.

When you use the name of a JVM profile anywhere in CICS, you must enter it using the same combination of upper and lower case characters that is present in the z/OS UNIX file name.

JVMPROGRAM (*data-value*)

requests statistics for a Java program that runs in a JVM. The data-value is the name of the program's PROGRAM resource definition.

LASTRESET (*data-area*)

returns a 4-byte packed decimal field giving the time at which the counters for the requested statistics were last reset. This is usually the time of the expiry of the last interval. The last reset time is always returned in local time.

There are two formats for the reset time:

- A composite (packed decimal format 0hhmmss+), which you obtain by using the LASTRESET option.
- Separate hours, minutes, and seconds, which you obtain by specifying the LASTRESETHRS, LASTRESETMIN, and LASTRESETSEC options respectively.

LASTRESETHRS (*data-area*)

returns a fullword binary field giving the hours component of the time at which the counters for the requested statistics were last reset (see the LASTRESET option).

LASTRESETMIN (*data-area*)

returns a fullword binary field giving the minutes component of the time at which the counters for the requested statistics were last reset (see the LASTRESET option).

LASTRESETSEC (*data-area*)

returns a fullword binary field giving the seconds component of the time at which the counters for the requested statistics were last reset (see the LASTRESET option).

LSRPOOL (*data-value*)

requests statistics on a VSAM LSR pool; data-value is the pool number, in the range 1-8, in fullword binary form.

MONITOR (*data-value*)

requests performance class statistics for a task when a data-value is supplied. The data-value is the task number, in 4-byte packed decimal format. Without a data-value, MONITOR requests global performance class statistics.

The monitoring performance class must be active for any statistics to be returned. If performance class is not active, the NOTFND condition is returned. For background information about monitoring, see the *CICS Performance Guide*.

MVSTCB(*data-value*)

requests statistics for an MVS TCB when a data-value is supplied. The data-value is the address of an MVS TCB. Without a data-value, MVSTCB requests global statistics for MVS TCBs in the CICS address space.

POOL(*data-value*)

requests statistics for a FEPI pool; data-value is the 8-character name of the pool.

POOL(*data-value*) **TARGET**(*data-value*)

requests statistics for a FEPI target within a FEPI pool. The POOL data-value identifies the pool, and the TARGET data-value identifies the system within the pool for which statistics are requested.

PROGAUTO

requests global statistics on the autoinstalled program definitions.

PROGRAM(*data-value*)

requests statistics for a program when a data-value is supplied. The data-value is the 8-character name of the program PROGRAM definition. Without a data-value, PROGRAM requests the global program statistics. CICS does not collect statistics for programs that run in a JVM when a COLLECT STATISTICS PROGRAM command is issued; you must use the COLLECT STATISTICS JVMPROGRAM command to obtain these statistics.

RECOVERY

requests global statistics on the recovery manager.

REQUESTMODEL(*data-value*)

requests statistics for a request model; data-value is the 8-character name of the request model resource definition.

SET(*ptr-ref*)

specifies a pointer reference to be set to the address of the data area containing the returned statistics. The first 2 bytes of the data area contain the length of the data area in halfword binary form.

STATS

requests global statistics on the statistics domain.

STORAGE(*data-value*)

requests statistics for a storage domain subpool when a data-value is present. The data-value is the 8-character name of a storage domain subpool. A complete list of the possible subpool names is documented in the *CICS Performance Guide*. Without a data-value, this option requests the global statistics for the CICS dynamic storage areas.

STREAMNAME(*data-value*)

requests statistics for a logstream when a data-value is supplied. The data-value is the 26-character name of the logstream. Without a data-value, STREAMNAME requests the global statistics for the CICS log manager.

SUBPOOL(*data-value*)

requests statistics for a storage manager domain subpool. The data-value is the 8-character name of a domain subpool. For tables of the CICS storage manager domain subpools, see the *CICS Performance Guide*.

SYSDUMPCODE (*data-value*)

requests statistics for a system dump code when a data-value is supplied. The data-value is the 8-character dump code. Without a data-value, SYSDUMPCODE requests global statistics on system dumps.

TABLEMGR

requests global statistics on the table manager.

TARGET (*data-value*) **NODE** (*data-value*)

requests statistics for a FEPI connection. The NODE data-value is the 8-character name of the terminal which FEPI simulates, and the TARGET data-value is the 8-character name of the system to which FEPI appears as a secondary logical unit.

TASKSUBPOOL

requests global statistics for the storage manager task subpools.

TCLASS (*data-value*)

requests statistics for a transaction class; data-value is the class number, in the range 1-10, in fullword binary form. Transaction classes are no longer identified by number, but instead by an 8-character identifier.

When you use the TCLASS option to request statistics for a class (as opposed to TRANCLASS), a conversion from fullword binary number to 8-character value is made on your behalf (for example, TCLASS(01) becomes the equivalent of TRANCLASS('DFHTCL01')).

TCPIP

requests global statistics for IP sockets.

TCPIPSERVICE (*data-value*)

requests the statistics for a TCP/IP service; data-value is the 8-character name of the TCP/IP service.

TDQUEUE (*data-value*)

requests statistics for a transient data queue when data-value is supplied. The data-value is the 4-character name of the queue. Without a data-value, TDQUEUE requests the global statistics for transient data.

TERMINAL (*data-value*)

requests statistics for a terminal; data-value is the 4-character terminal identifier (from the TERMINAL definition).

TRANCLASS (*data-value*)

requests statistics for a transaction class; data-value is the 8-character name of the class from the TRANCLASS definition.

TRANDUMPCODE (*data-value*)

requests statistics for a transaction dump code when a data-value is supplied. The data-value is the 4-character dump code. Without a data-value, TRANDUMPCODE requests global statistics on transaction dumps.

TRANSACTION (*data-value*)

requests statistics for a transaction when a data-value is supplied. The data-value is the 4-character transaction identifier (from the TRANSACTION definition). Without a data-value, TRANSACTION requests global statistics on transactions.

TSQUEUE

requests global statistics on temporary storage.

VTAM (now **z/OS Communications Server**)

requests global statistics on z/OS Communications Server.

Conditions

INVREQ

RESP2 values:

- 4 The TCLASS value was not in the range 1-10, or the LSRPOOL value was not in the range 1-8.

IOERR

RESP2 values:

- 3 The requested statistics area was not functioning. This happens if, for instance, statistics control blocks are overwritten.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

NOTFND

RESP2 values:

- 1 The requested resource cannot be found (for example, if the JOURNALNAME you specify is not known to CICS).
- 2 The type of resource is not defined in the CICS system (for example, FEPI statistics are requested with POOL or NODE when the FEPI system initialization parameter specifies NO).

Note: NOTFND is also returned if you attempt to use COLLECT STATISTICS PROGRAM when you should use COLLECT STATISTICS JVMPROGRAM. See “Collecting statistics for Java programs that run in a JVM” for additional information.

Examples

CICS provides a sample COLLECT STATISTICS application (DFH0STAT) that uses virtually all the options described in this section. This set of programs illustrates ways of using the COLLECT STATISTICS and INQUIRE commands to produce information about a CICS system. The reports include a CICS and MVS storage analysis that can be used as an aid to specifying the DSA LIMIT parameters.

See the *CICS Performance Guide* for information about installing and operating the DFH0STAT application. The source code for the application can be found in CICSTS42.CICS.SDFHSAMP.

Collecting statistics for Java programs that run in a JVM

Statistics for Java programs that run in a JVM are collected separately from statistics for other programs, because the JVM programs are not loaded by CICS.

CICS does not collect statistics for JVM programs when a COLLECT or PERFORM STATISTICS PROGRAM command is issued; to see them, you need to use the COLLECT or PERFORM STATISTICS JVMPROGRAM command instead.

However, when you browse program names using the INQUIRE PROGRAM command, JVM programs are found. If you have an application that collects

statistics for programs by browsing with the INQUIRE PROGRAM command, and then issuing the COLLECT STATISTICS PROGRAM command for the program names that it finds, the application will now receive a "not found" response when it attempts to collect statistics for any JVM programs. (Previously, zeros would have been returned as statistics values for these programs.)

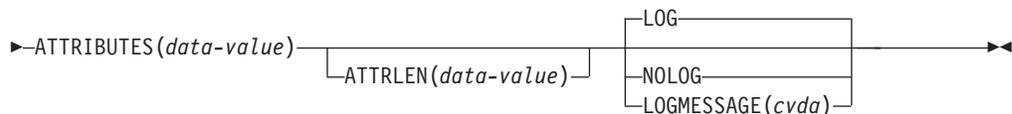
To avoid receiving this response, make the application check the RUNTIME value for each program name that it finds. If the RUNTIME value is JVM, the application should not issue the COLLECT STATISTICS PROGRAM command for that program name. If you want to see the statistics for programs with a RUNTIME value of JVM, you can make the application issue the COLLECT STATISTICS JVMPROGRAM command for those programs. Note that the statistics information that is collected for JVM programs is not the same as that collected for other programs.

CREATE ATOMSERVICE

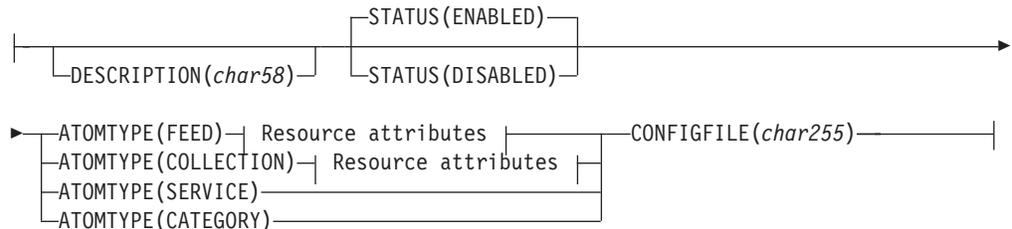
Define an ATOMSERVICE resource definition in the local CICS region.

CREATE ATOMSERVICE

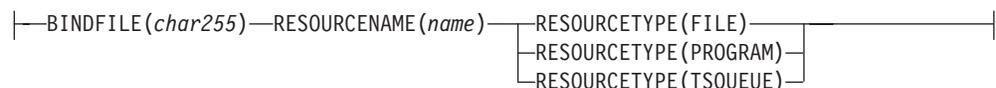
►► CREATE ATOMSERVICE(*data-value*)



CREATE ATOMSERVICE attribute values:



Attributes for FEED or COLLECTION:



Conditions: INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use ATTRIBUTES(*data-area*) instead of ATTRIBUTES(*data-value*).

Description

The CREATE ATOMSERVICE command builds an ATOMSERVICE definition. It does not use a resource definition stored in the CSD. If an ATOMSERVICE definition already exists with the name that you specify in the local CICS region, the command fails unless the existing ATOMSERVICE definition is disabled, in which case the new definition replaces the old one. If no ATOMSERVICE definition with the name specified exists, the new definition is added.

A syncpoint is implicit in CREATE ATOMSERVICE processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE command is successful, and rolled back if not.

Options

ATTRIBUTES(*data-value*)

Specifies the attributes of the ATOMSERVICE definition being added. The list of attributes must be coded as a single character string using the syntax shown in **ATOMSERVICE definition attributes**. See the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN(*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length must not exceed 32767 bytes.

LOGMESSAGE(*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

ATOMSERVICE(*data-value*)

Specifies the 8-character name of the ATOMSERVICE definition to be added to the CICS region.

Conditions

INVREQ

RESP2 values:

- n** The ATTRIBUTES string contains a syntax error, or an error occurred during either the discard or resource definition phase of the processing. The RESP2 value is associated with a message written to the transient data queue CSMT, which identifies more precisely the nature of the error.
- 7** The LOGMESSAGE cvda value is not valid.
- 200** The command ran in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1** The length that you have specified in ATTRLEN is negative.

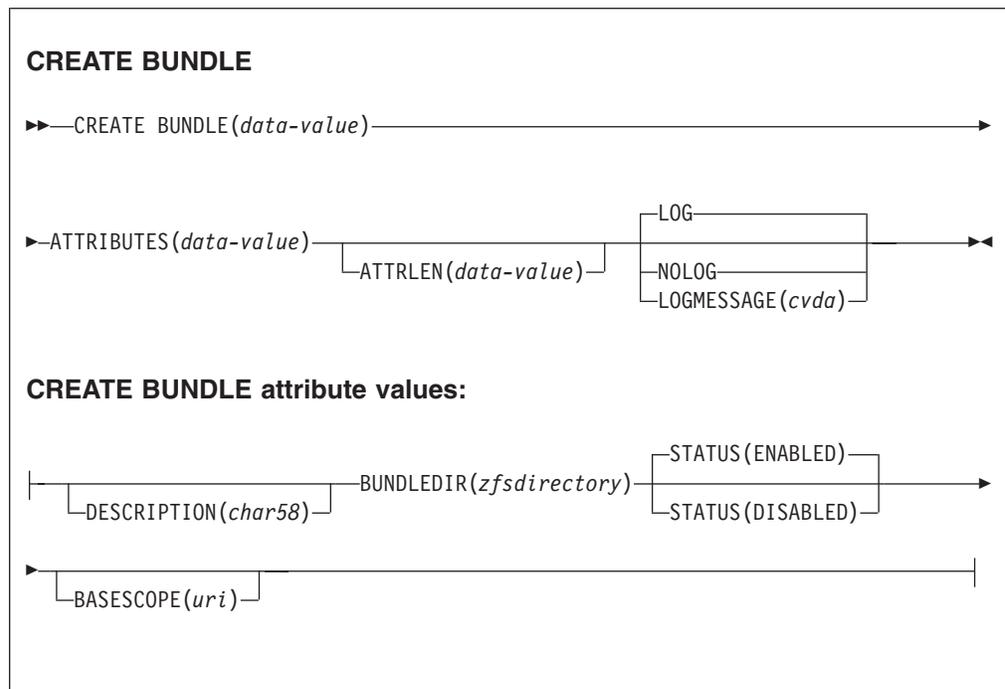
NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

CREATE BUNDLE

Define a BUNDLE resource in the local CICS region.



Conditions: INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use ATTRIBUTES(*data-area*) instead of ATTRIBUTES(*data-value*).

Description

The CREATE BUNDLE command installs a BUNDLE definition with the attributes specified on the command. It does not use a resource definition stored in the CSD. If there is already a BUNDLE with the name that you specify in the local CICS region, and the existing BUNDLE is disabled, the new definition replaces the old one; if an existing BUNDLE is not disabled, the CREATE command fails.

Options

ATTRIBUTES(*data-value*)

Specifies the attributes of the BUNDLE being added. The list of attributes must be coded as a single character string using the syntax shown in **BUNDLE attributes**. See the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN(*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length must not exceed 32 767 bytes.

BUNDLE(*data-value*)

Specifies the 8-character name of the BUNDLE definition to be added to the CICS region.

LOGMESSAGE(*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

Conditions**INVREQ**

RESP2 values:

- n** The ATTRIBUTES string contains a syntax error, or an error occurred during either the discard or resource definition phase of the processing. The RESP2 value is associated with a message written to the transient data queue CSMT, which identifies more precisely the nature of the error.
- 612** Installation of the BUNDLE definition failed because it already exists.
- 632** Installation of BUNDLE resource *resource* failed because the manifest is not valid.
- 633** Installation of BUNDLE resource *resource* failed because it had no manifest.
- 686** Installation of the BUNDLE resource failed because an unexpected resource error occurred.

LENGERR

RESP2 values:

- 1** The length that you have specified in ATTRLEN is negative.

NOTAUTH

RESP2 values:

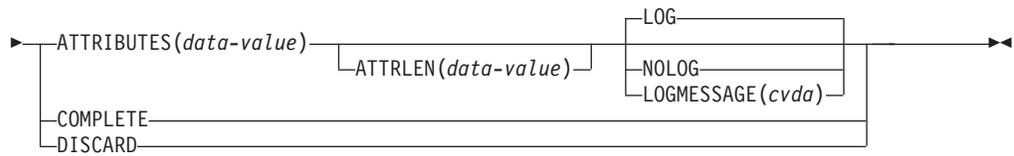
- 100** The user associated with the issuing task is not authorized to use this command.
- 101** The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

CREATE CONNECTION

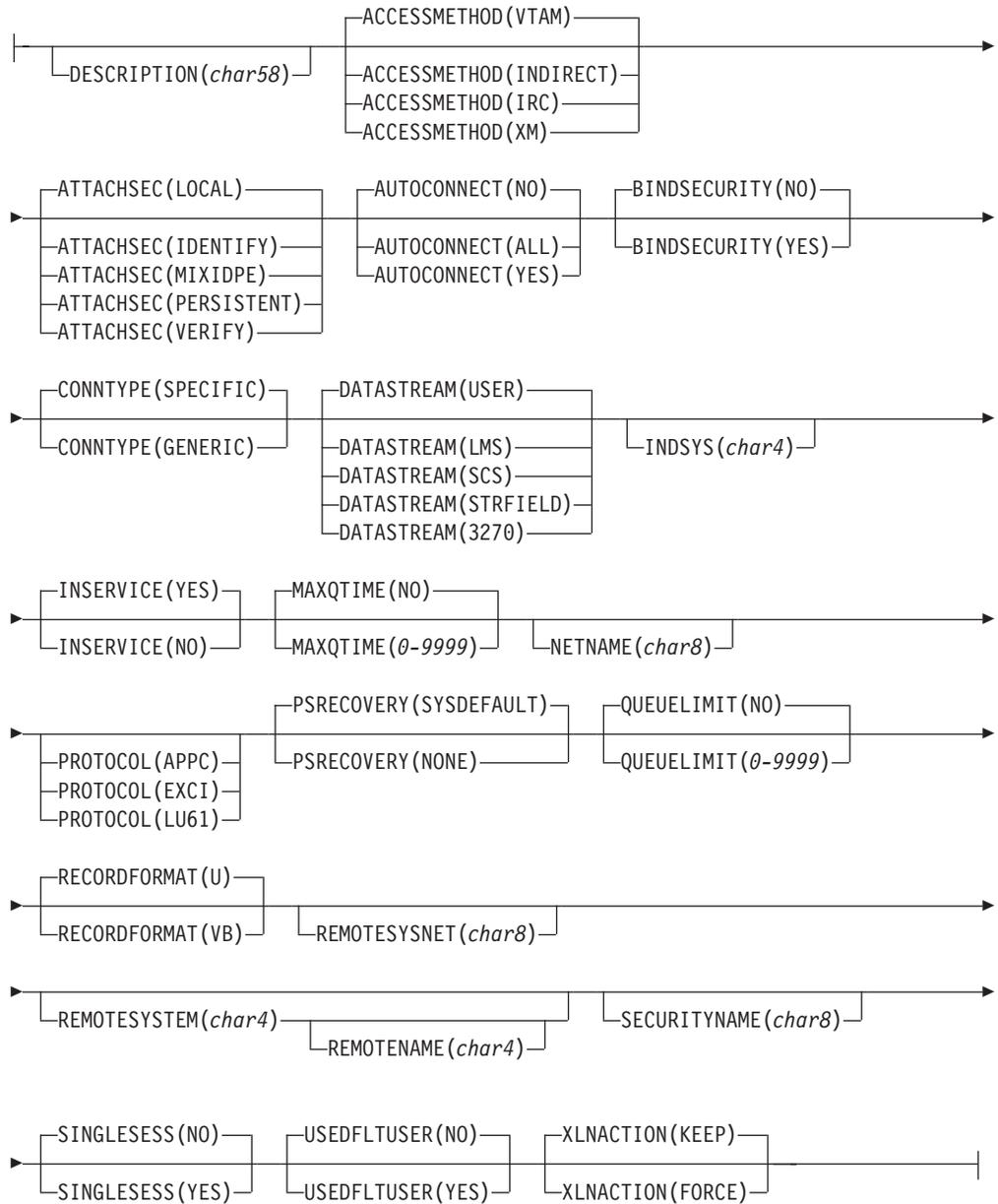
Define a CONNECTION in the local CICS region.

CREATE CONNECTION

►►—CREATE CONNECTION(*data-value*)—————►



CREATE CONNECTION attribute values:



Note: VTAM is now z/OS Communications Server.

Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use `ATTRIBUTES (data-area)` instead of `ATTRIBUTES (data-value)`.

Description

The CREATE CONNECTION commands, in combination with the CREATE SESSIONS commands, install CONNECTION and SESSIONS definitions with the attribute specified on the command to the local CICS region. They do not use resource definitions stored in the CSD. See “Creating resource definitions” on page 24 for other general rules about CREATE commands.

Note: CREATE CONNECTION creates an MRO, APPC, or LUTYPE6.1 communication link to a remote system. See also CREATE IPCONN in CICS System Programming Reference. Like a CONNECTION, an IPCONN defines a communication link to a remote system, but in this case the connection uses the TCP/IP protocol.

To create a new CONNECTION, you issue a series of commands in this order:

1. CREATE CONNECTION with the ATTRIBUTES and ATTRLEN options
2. CREATE SESSIONS
3. Additional CREATE SESSIONS if desired (only one group of sessions is required, but you can define additional groups)
4. CREATE CONNECTION with the COMPLETE option.

The CONNECTION is not added until all of these steps take place. During the time the definition is being built (that is, between the initial and final CREATE CONNECTIONS), you may not:

- Define other resources of any type, including other connections
- Issue a SYNCPOINT (or any command that implies one)
- Terminate your task (normally)

However, if you encounter an error or problem during the course of building a CONNECTION definition, you can terminate the process at any point by issuing a CREATE CONNECTION DISCARD command. If you do this, CICS discards the partial CONNECTION definition and any SESSIONS created for it.

Otherwise, when the final CREATE CONNECTION COMPLETE command is issued, CICS adds the CONNECTION and its SESSIONS to its resource definitions, replacing a CONNECTION definition of the same name if one exists.

CICS also performs an implicit SYNCPOINT command during the processing of the final CREATE for a connection, unless it contains an error that can be detected early in the processing. The syncpoint commits uncommitted changes to recoverable resources made up to that point in the task if the definition is successful, and rolls back changes, as if SYNCPOINT ROLLBACK had been issued, if the definition fails or ends in a DISCARD. See “Creating resource definitions” on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES (*data-value*)

specifies the attributes of the CONNECTION being added. The list of attributes must be coded as a single character string using the syntax shown in **CONNECTION attributes**. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and CONNECTION definition attributes for details about specific attributes.

Note: You can assign default values for all attributes of a CONNECTION definition by specifying an ATTRLEN value of 0. You still need to specify the ATTRIBUTES option, however, even though its value is not used.

ATTRLEN (*data-value*)

specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length can be from 0 to 32767.

COMPLETE

specifies that the set of definitions for this CONNECTION is complete and should be added to the CICS system.

CONNECTION (*data-value*)

specifies the 4-character name of the CONNECTION definition to be added.

DISCARD

specifies that the CONNECTION definition under construction is not to be completed and that it and any SESSIONS created for it are to be discarded and *not* added.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, "RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands," on page 837 for information on RESP2 values.
- 7 The LOGMESSAGE cvda value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length specified in ATTRLEN is negative.

NOTAUTH

RESP2 values:

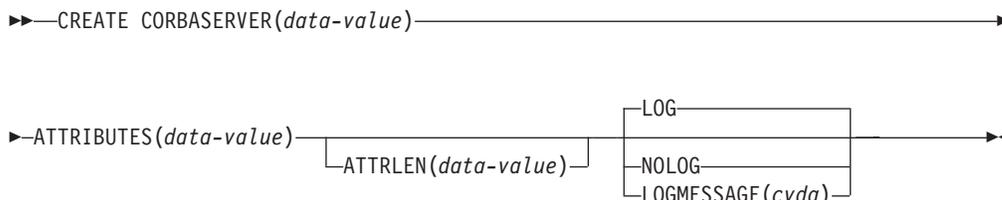
- 100 The user associated with the issuing task is not authorized to use this command.

102 The user associated with the task issuing the CREATE CONNECTION command is not an authorized surrogate of the user specified in SECURITYNAME.

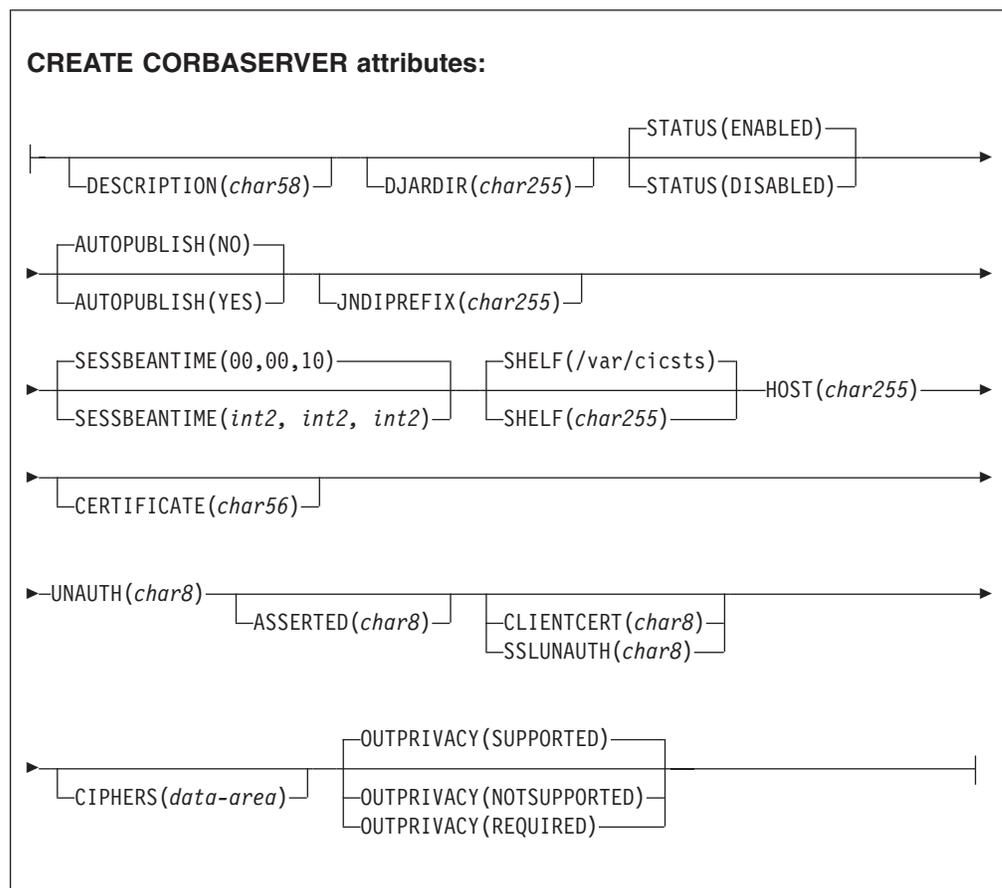
CREATE CORBASERVER

Define a CorbaServer in the local CICS region.

CREATE CORBASERVER



Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH



Description

CREATE CORBASERVER commands build CORBASERVER definitions, without reference to data stored on the CSD file. If the named CorbaServer already exists, an error condition (INVREQ, RESP2=612) is raised.

A syncpoint is implicit in CREATE CORBASERVER processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not. See “Creating resource definitions” on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES (*data-value*)

specifies the attributes of the CORBASERVER being added. The list of attributes must be coded as a single character string using the syntax shown in **CORBASERVER attributes**. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

Note: You can assign default values for all attributes of a CORBASERVER definition by specifying an ATTRLEN value of 0. You still need to specify the ATTRIBUTES option, however, even though its value is not used.

ATTRLEN (*data-value*)

specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length can be from 0 to 32767.

CORBASERVER (*data-value*)

specifies the 4-character name of the CorbaServer definition to be added.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n** There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. The RESP2 value is associated with a message written to the transient data queue CSMT which identifies more precisely the nature of the error. See Appendix C, “RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands,” on page 837 for information on RESP2 values.
- 200** The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.
- 612** Installation of the CorbaServer definition failed because it already exists.

- 613 Installation of the CorbaServer definition failed because an invalid JNDIprefix, shelf or certificate value has been specified.
- 615 Installation of the CorbaServer definition failed because the resource resolution transaction CEJR could not attach.
- 616 Installation of the CorbaServer defintion failed because the specified certificate is not known to the external security manager (ESM).
- 619 Installation of the CorbaServer defintion failed because one of its associated TCPIP SERVICES was not installed.

LENGERR

RESP2 values:

- 1 The length specified in ATTRLEN is negative.

NOTAUTH

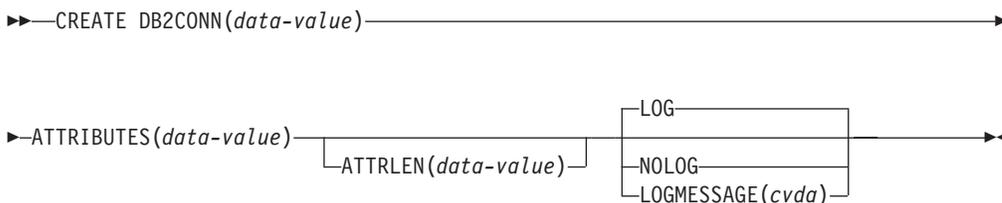
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

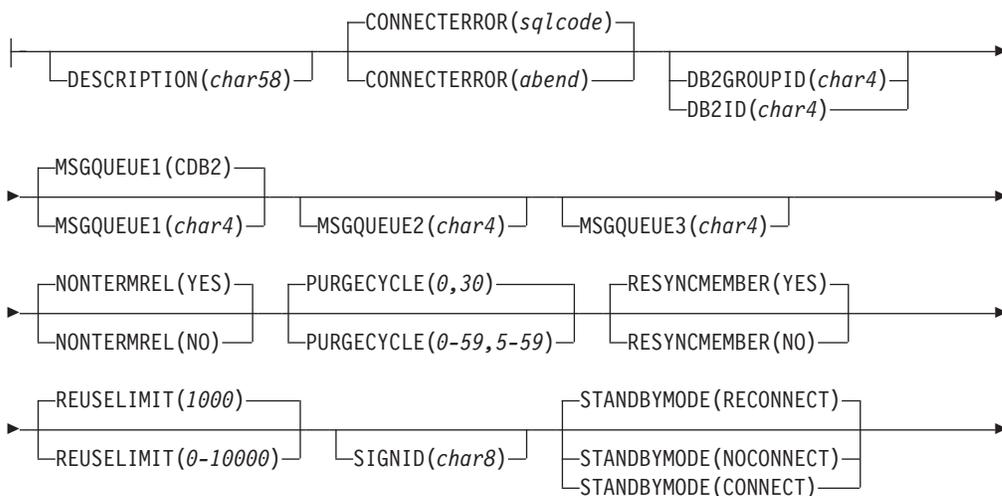
CREATE DB2CONN

Define a DB2CONN in the local system.

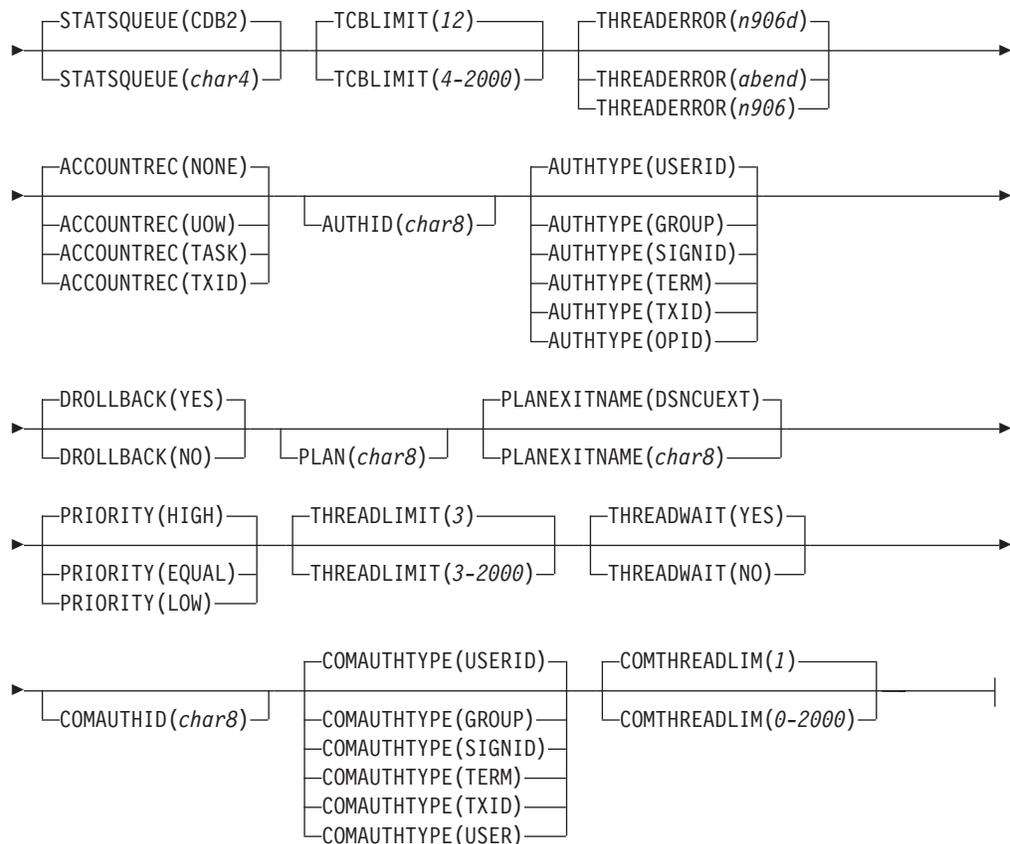
CREATE DB2CONN



CREATE DB2CONN attribute values:



I



Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use **ATTRIBUTES**(data-area) instead of **ATTRIBUTES**(data-value).

Description

The CREATE DB2CONN command installs a DB2CONN definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is already a DB2CONN in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

A sync point is implicit in CREATE DB2CONN processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not.

Options

ATTRIBUTES(data-value)

Specifies the attributes of the DB2CONN that is being added. The list of attributes must be coded as a single character string using the syntax shown in **DB2CONN attributes**. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

Note: You can assign default values for all attributes of a DB2CONN definition by specifying an ATTRLEN value of 0. You still need to specify the ATTRIBUTES option, however, even though its value is not used.

ATTRLEN (*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length can be from 0 to 32767.

DB2CONN (*data-value*)

Specifies the 8-character name of the DB2CONN definition to be added to the CICS region.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n A syntax error occurred in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, "RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands," on page 837 for information about RESP2 values.
- 7 The LOGMESSAGE CVDA value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

NOTAUTH

RESP2 values:

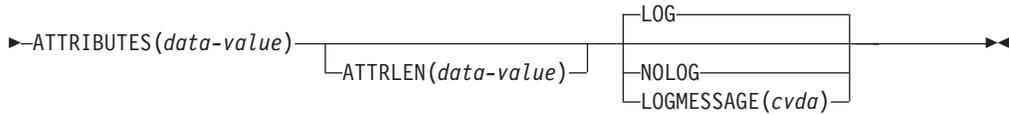
- 100 The user associated with the issuing task is not authorized to use this command.
- 102 The user associated with the issuing task is not an authorized surrogate of the user specified in the AUTHID, COMAUTHID, or SIGNID parameter.
- 103 The user associated with the issuing task is not authorized to create a DB2CONN with an AUTHTYPE or COMAUTHTYPE parameter.

CREATE DB2ENTRY

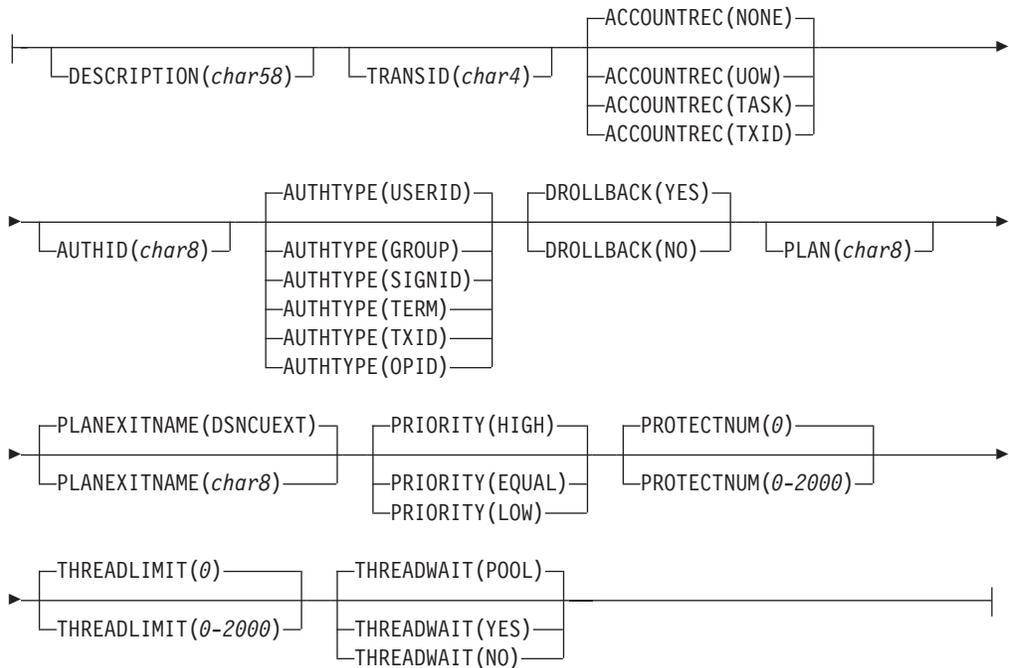
Define a DB2ENTRY in the local system.

CREATE DB2ENTRY

►► CREATE DB2ENTRY (*data-value*)



CREATE DB2ENTRY attribute values:



Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use ATTRIBUTES(*data-area*) instead of ATTRIBUTES(*data-value*).

Description

The CREATE DB2ENTRY command installs a DB2ENTRY definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is already a DB2ENTRY with the name you specify in the local CICS region, the command fails unless the existing DB2ENTRY is disabled, in which case the new definition replaces the old one. If no DB2ENTRY with the name specified exists, the new definition is added.

A syncpoint is implicit in CREATE DB2ENTRY processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed

if the CREATE executes successfully, and rolled back if not.

Options

ATTRIBUTES (*data-value*)

Specifies the attributes of the DB2ENTRY being added. The list of attributes must be coded as a single character string using the syntax shown in **DB2ENTRY attributes**. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and DB2ENTRY resources in the Resource Definition Guide for details about specific attributes.

Note: You can assign default values for all attributes of a DB2ENTRY definition by specifying an ATTRLEN value of 0. You still need to specify the ATTRIBUTES option, however, even though its value is not used.

ATTRLEN (*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length can be from 0 to 32767.

DB2ENTRY (*data-value*)

Specifies the 8-character name of the DB2ENTRY definition to be added to the CICS region.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, “RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands,” on page 837 for information on RESP2 values.
- 7 The LOGMESSAGE cvda value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

NOTAUTH

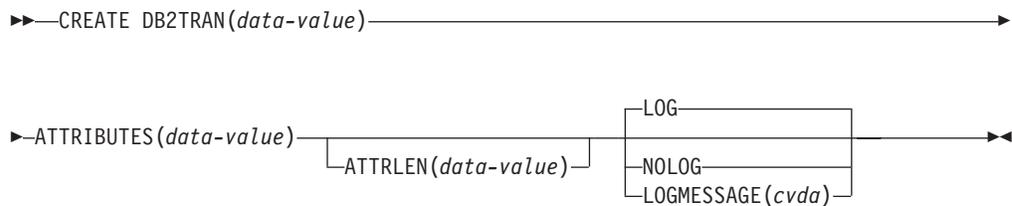
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to create a DB2ENTRY definition with this name.
- 102 The user associated with the issuing task is not an authorized user specified in the AUTHID parameter.
- 103 The user associated with the issuing task is not authorized to create this DB2ENTRY with an AUTHTYPE parameter.

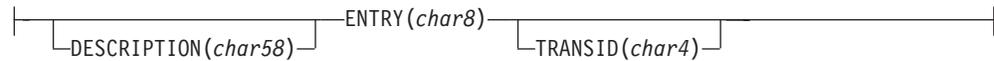
CREATE DB2TRAN

Define a DB2TRAN in the local system.

CREATE DB2TRAN



CREATE DB2TRAN attribute values:



Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use `ATTRIBUTES(data-area)` instead of `ATTRIBUTES(data-value)`.

Description

The CREATE DB2TRAN command installs a DB2TRAN definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is already a DB2TRAN with the name you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added. If there is already a DB2TRAN in the local CICS region that specifies the same TRANSID, the command fails, as each transaction can only have one DB2TRAN definition.

A syncpoint is implicit in CREATE DB2TRAN processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not.

Options

ATTRIBUTES(data-value)

specifies the attributes of the DB2TRAN being added. The list of attributes must be coded as a single character string using the syntax shown in

DB2TRAN attributes. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

Note: You can assign default values for all attributes of a DB2TRAN definition by specifying an ATTRLEN value of 0. You still need to specify the ATTRIBUTES option, however, even though its value is not used.

ATTRLEN(*data-value*)

specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length can be from 0 to 32767.

DB2TRAN(*data-value*)

specifies the 8-character name of the DB2TRAN definition to be added to the CICS region.

LOGMESSAGE(*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, “RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands,” on page 837 for information on RESP2 values.
- 7 The LOGMESSAGE cvda value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

NOTAUTH

RESP2 values:

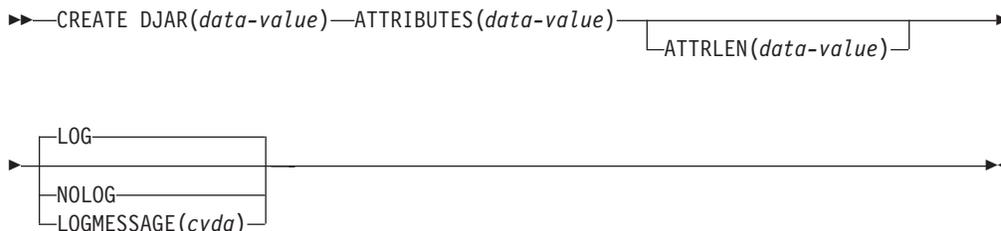
- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to create a DB2TRAN definition and associate it with the names DB2ENTRY.
- 102 The user associated with the issuing task is not an authorized surrogate of the user specified in the AUTHID parameter of the DB2ENTRY named in the DB2TRAN.

103 The user associated with the issuing task is not authorized to associate this DB2TRAN with the names DB2ENTRY specifying AUTHTYPE.

CREATE DJAR

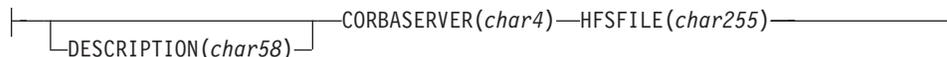
Define a deployed JAR file in the local CICS region.

CREATE DJAR



Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

CREATE DJAR attributes:



Description

The CREATE DJAR command installs a DJAR definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If the named DJAR already exists, it is replaced by the new definition and any beans installed in the DJAR are discarded. Any new beans in the new DJAR are installed.

A syncpoint is implicit in CREATE DJAR processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not. See "Creating resource definitions" on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES(data-value)

specifies the attributes of the CorbaServer being added. The list of attributes must be coded as a single character string using the syntax shown in **DJAR attributes**. See "The ATTRIBUTES option" on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

Note: You can assign default values for all attributes of a DJAR definition by specifying an ATTRLEN value of 0. You still need to specify the ATTRIBUTES option, however, even though its value is not used.

ATTRLEN(*data-value*)

specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length can be from 0 to 32767.

DJAR(*data-value*)

specifies the 8-character name of the DJAR definition to be added.

Note: The names of *static* DJAR definitions (those installed from a CSD or by means of CREATE DJAR) are limited to 8 characters. The names of *dynamic* DJAR definitions (those installed by the CICS scanning mechanism) can be up to 32 characters long.

LOGMESSAGE(*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

Conditions**ILLOGIC**

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. The RESP2 value is associated with a message written to the transient data queue CSMT which identifies more precisely the nature of the error. See Appendix C, "RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands," on page 837 for information on RESP2 values.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.
- 612 Installation of the DJAR definition failed because it already exists.
- 613 Installation of the DJAR definition failed because HFSFILE, state, CorbaServer, or DJAR invalid.
- 614 Installation of the DJAR definition failed because specified CORBASERVER does not exist.
- 615 Installation of the DJAR definition failed because the resource resolution transaction CEJR could not attach.
- 617 Installation of the DJAR definition failed because specified CORBASERVER is not in a valid state.
- 618 Installation of a DJAR definition failed because its HFSFILE is a duplicate of one that already exists.

LENGERR

RESP2 values:

- 1 The length specified in ATTRLEN is negative.

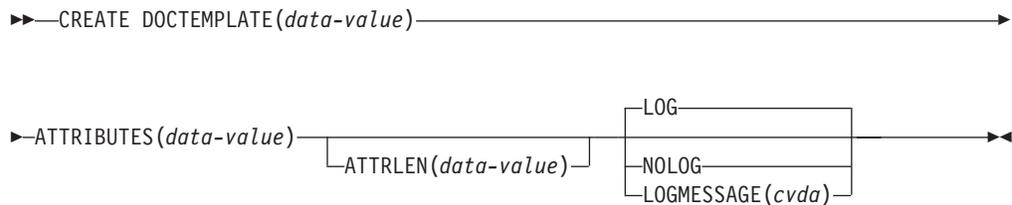
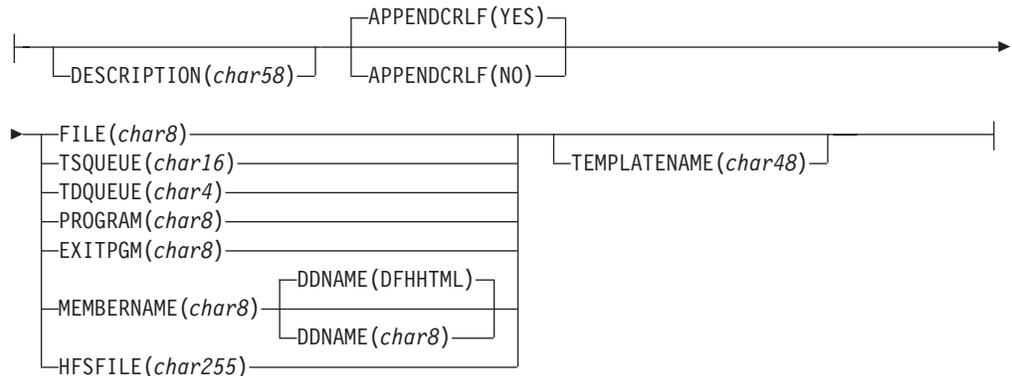
NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

CREATE DOCTEMPLATE

Define a document template.

CREATE DOCTEMPLATE**CREATE DOCTEMPLATE attribute values:****Conditions:** INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use **ATTRIBUTES(data-area)** instead of **ATTRIBUTES(data-value)**.

Description

The CREATE DOCTEMPLATE command installs a DOCTEMPLATE definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is already a document template with the name you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

A syncpoint is implicit in CREATE DOCTEMPLATE processing, except when an exception condition is detected early in processing the command. Uncommitted

changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not. See “Creating resource definitions” on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES (*data-value*)

specifies the attributes of the DOCTEMPLATE being added. The list of attributes must be coded as a single character string using the syntax shown in **DOCTEMPLATE attributes**. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN (*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length must not exceed 32 767 bytes.

DOCTEMPLATE (*data-value*)

Specifies the 8-character name of the DOCTEMPLATE definition to be added to the CICS region.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

Conditions

INVREQ

RESP2 values:

- n** There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, “RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands,” on page 837 for information on RESP2 values.
- 7** The LOGMESSAGE cvda value is not valid.
- 200** The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET for a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1** The length you have specified in ATTRLEN is negative.

NOTAUTH

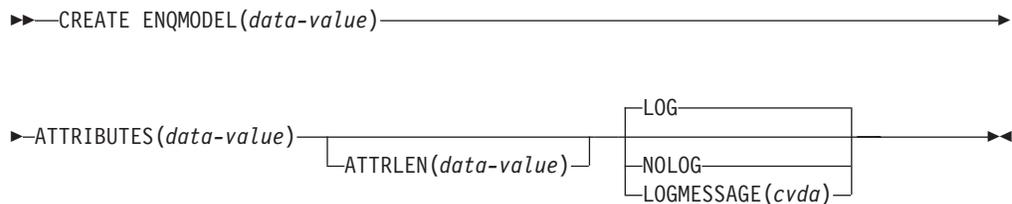
RESP2 values:

- 100** The user associated with the issuing task is not authorized to use this command.
- 101** The user associated with the issuing task is not authorized to create a DOCTEMPLATE resource definition with this name.

CREATE ENQMODEL

Define an ENQMODEL resource definition.

CREATE ENQMODEL



CREATE ENQMODEL attribute values:



Conditions: INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use **ATTRIBUTES**(*data-area*) instead of **ATTRIBUTES**(*data-value*).

Description

The CREATE ENQMODEL command installs a ENQMODEL definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is already an ENQMODEL with the name you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

When CREATE is issued, the ENQMODEL is put into the WAITING state until there are no enqueues in the local system which match the ENQNAME pattern. It is then ENABLED or DISABLED, as specified in the CREATE command.

ENQMODELs forming nested generic enqnames must either be installed in the disabled state, or be installed in order, from the most to the least specific. If another ENQMODEL with the same or a less specific nested enqname is already installed enabled, INVREQ is returned to the caller.

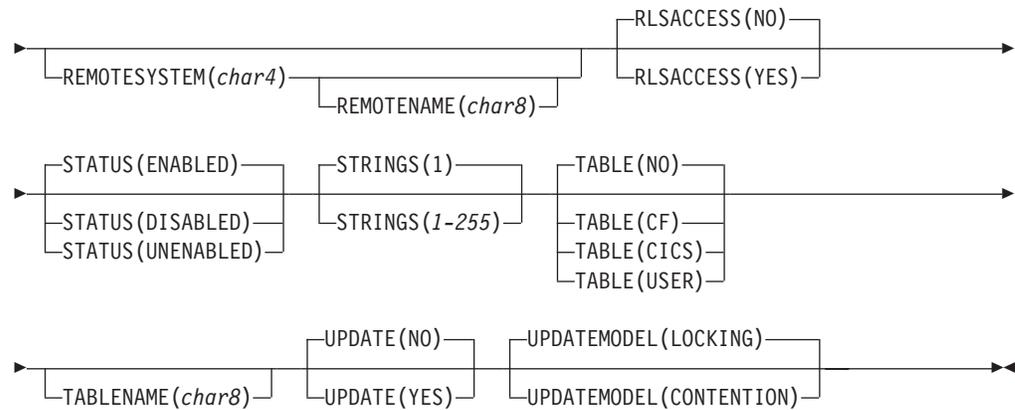
For example: If an ENQMODEL containing AB* is installed, it must be discarded or disabled before creating an ENQMODEL with ABCD*.

A syncpoint is implicit in CREATE ENQMODEL processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not. See "Creating resource definitions" on page 24 for other general rules governing CREATE commands

Options

ATTRIBUTES(*data-value*)

Specifies the attributes of the ENQMODEL being added. The list of attributes must be coded as a single character string using the syntax shown in



Note to COBOL programmers: In the syntax above, you must use `ATTRIBUTES(data-area)` instead of `ATTRIBUTES(data-value)`.

Description

The CREATE FILE command installs a FILE definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is already a file with the name you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

A sync point is implicit in CREATE FILE processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not. See “Creating resource definitions” on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES(*data-value*)

Specifies the attributes of the file that is added. The list of attributes must be coded as a single character string using the syntax shown. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and File attributes the *CICS Resource Definition Guide* for details about specific attributes.

Note: You can assign default values for all attributes of a FILE definition by specifying an ATTRLEN value of 0. You still must specify the ATTRIBUTES option, however, even though its value is not used.

ATTRLEN(*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length can be from 0 to 32767.

FILE(*data-value*)

Specifies the 8-character name of the FILE definition to be added to the CICS region.

LOGMESSAGE(*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

Conditions**ILLOGIC**

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, "RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands," on page 837 for information about RESP2 values.
- 7 The LOGMESSAGE cvda value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to create a FILE definition with this name.

CREATE IPCONN

Define and install an IPCONN in the local CICS region.

CREATE IPCONN

▶ CREATE IPCONN(*data-value*)

▶ ATTRIBUTES(*data-value*)

- ATTRLEN(*data-value*)
- LOG
- NOLOG
- LOGMESSAGE(*cvda*)

CREATE IPCONN attribute values:

DESCRIPTION(*char58*)

- APPLID(*IPCONN name*)
- APPLID(*applid*)
- AUTOCONNECT(NO)
- AUTOCONNECT(YES)

HOST(*name*)

- INSERVICE(YES)
- INSERVICE(NO)

LINKAUTH(SECUSER)

- SECURITYNAME(*name*)
- LINKAUTH(CERTUSER)
- MAXQTIME(NO)
- MAXQTIME(*seconds*)

MIRRORLIFE(REQUEST)

- MIRRORLIFE(TASK)
- MIRRORLIFE(UOW)
- NETWORKID(*networkID*)
- PORT(NO)
- PORT(*number*)

QUEUELIMIT(NO)

- QUEUELIMIT(*number*)
- RECEIVECOUNT(1)
- RECEIVECOUNT(*number*)
- SENDCOUNT(1)
- SENDCOUNT(*number*)

TCPIPSERVICE(*name*)

- XLNACTION(KEEP)
- XLNACTION(FORCE)

SSL(NO)

- SSL(YES)
- CERTIFICATE(*label*)
- CIPHERS(*value*)

USERAUTH(LOCAL)

- USERAUTH(DEFAULTUSER)
- USERAUTH(IDENTIFY)
- USERAUTH(VERIFY)

I

Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use `ATTRIBUTES(data-area)` instead of `ATTRIBUTES(data-value)`.

Description

The CREATE IPCONN command installs an IPCONN definition with the attributes specified on the command. It does not use a resource definition stored in the CSD. If there is already an IPCONN with the name you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

Note: CREATE IPCONN creates a TCP/IP communication link to a remote system. See also “CREATE CONNECTION” on page 45. Like an IPCONN, a CONNECTION defines a communication link to a remote system, but in this case the connection uses the APPC or LUTYPE6.1 communication protocol (intersystem communication), or the IRC, XM, or XCF/MRO access method (multiregion operation).

Note that for connectivity to be achieved when you install the IPCONN definition:

1. The TCPIP SERVICE definition named on the TCPIP SERVICE option of this IPCONN definition must also be installed in this region and must specify PROTOCOL(IPIC).
2. Corresponding IPCONN and TCPIP SERVICE definitions must be installed in the remote region. Characteristics of corresponding IPCONN and TCPIP SERVICE definitions are as follows:
 - The HOST option of the IPCONN definition on the remote region must specify this region.
 - The PORT option of the IPCONN definition on the remote region must specify the same port number as that specified on the PORTNUMBER option of the local TCPIP SERVICE definition named by this IPCONN.
 - The TCPIP SERVICE definition on the remote region (named by the IPCONN definition on the remote region) must specify PROTOCOL(IPIC) and, on its PORTNUMBER option, the same port number as that specified by the PORT option of this IPCONN.

If this IPCONN is to be used for distributed program link (DPL) between CICS TS 3.2 or later regions, or transaction routing between CICS TS 4.1 or later regions, or function shipping file control, transient data, or temporary storage requests between CICS TS 4.2 or later regions, using IPIC connectivity, its name must match the 4-character *local name* (SYSID) by which CICS knows the remote system, padded with four trailing blanks.

Note: The name (SYSID) of the remote, target region, of a DPL request can be specified by using any of the following methods:

- The REMOTESYSTEM option of the installed PROGRAM definition
- The SYSID option of the EXEC CICS LINK PROGRAM command
- The dynamic routing program

For details of the attributes of IPCONN and TCPIP SERVICE definitions, see the *CICS Resource Definition Guide*. For guidance on defining IPIC connections, see the *CICS Intercommunication Guide*.

A sync point is implicit in CREATE IPCONN processing, except when an exception condition is detected early in processing the command. Uncommitted changes to

recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not.

See “Creating resource definitions” on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES(*data-value*)

Specifies the attributes of the IPCONN that is being added. The list of attributes must be coded as a single character string using the syntax shown in IPCONN attributes. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN(*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length must not exceed 32767 bytes.

IPCONN(*data-value*)

Specifies the 8-character name of the connection to the remote system (that is, the name of the IPCONN definition to be created).

LOGMESSAGE(*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, “RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands,” on page 837 for information on RESP2 values.
- 7 The LOGMESSAGE CVDA value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET, or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

NOTAUTH

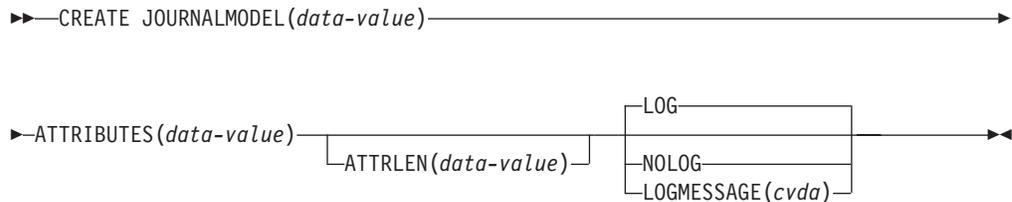
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 102 The user associated with the task issuing the CREATE IPCONN command is not an authorized surrogate of the user specified in the SECURITYNAME option.

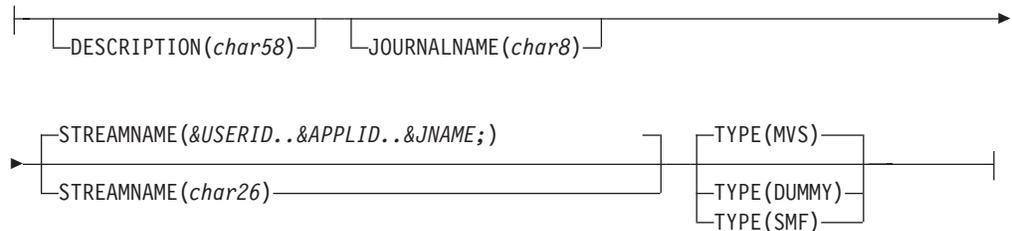
CREATE JOURNALMODEL

Define a journal model in the local CICS region.

CREATE JOURNALMODEL



CREATE JOURNALMODEL attribute values:



Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use `ATTRIBUTES(data-area)` instead of `ATTRIBUTES(data-value)`.

Description

The CREATE JOURNALMODEL command installs a JOURNALMODEL definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is already a journal model with the name you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

A syncpoint is implicit in CREATE JOURNALMODEL processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not. See "Creating resource definitions" on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES(data-value)

specifies the attributes of the JOURNALMODEL being added. The list of attributes must be coded as a single character string using the syntax shown in

JOURNALMODEL attributes. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

Note: You can assign default values for all attributes of a JOURNALMODEL definition by specifying an ATTRLEN value of 0. You still need to specify the ATTRIBUTES option, however, even though its value is not used.

ATTRLEN(*data-value*)

specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length can be from 0 to 32767.

JOURNALMODEL(*data-value*)

specifies the 8-character name of the JOURNALMODEL definition to be added to the CICS region.

LOGMESSAGE(*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, “RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands,” on page 837 for information on RESP2 values.
- 7 The LOGMESSAGE cvda value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

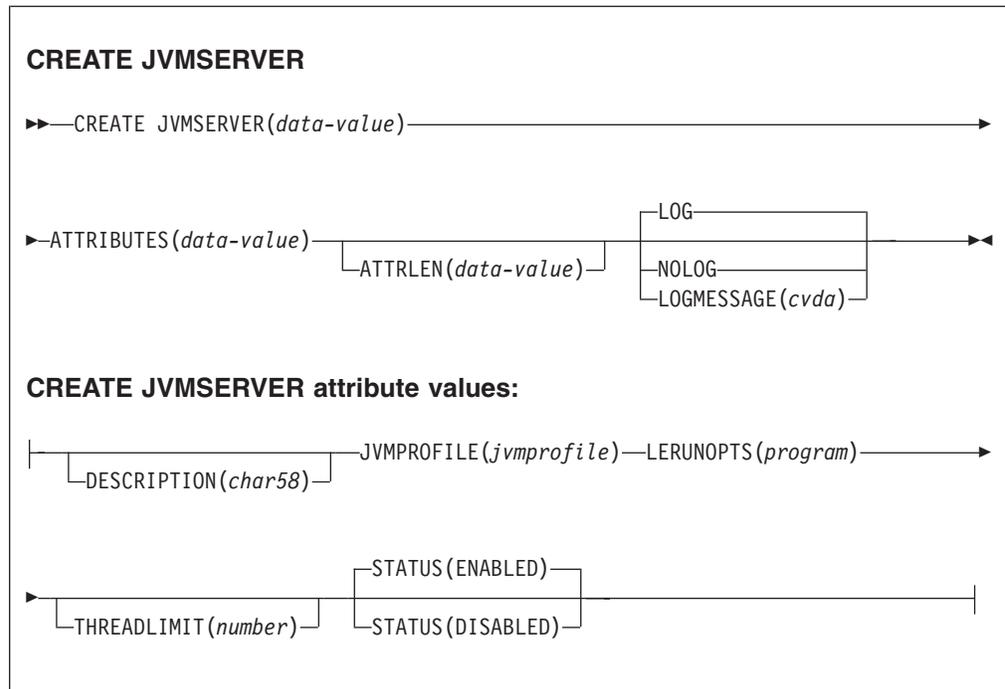
NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

CREATE JVMSERVER

Define a JVMSERVER resource in the local CICS region.



Conditions: INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use `ATTRIBUTES(data-area)` instead of `ATTRIBUTES(data-value)`.

Description

The **CREATE JVMSERVER** command installs a JVMSERVER definition with the attributes specified on the command. If a JVMSERVER with the name you specify in the local CICS region already exists, and the existing JVMSERVER resource is disabled, the new definition replaces the old one. If an existing JVMSERVER resource is not disabled, the CREATE command fails.

Options

ATTRIBUTES(*data-value*)

Specifies the attributes of the JVMSERVER resource. The list of attributes must be coded as a single character string using the syntax shown in **JVMSERVER attributes**.

See the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN(*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length must not exceed 32 767 bytes.

LOGMESSAGE(*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

JVMSEVER(*data-value*)

Specifies the 8-character name of the JVMSEVER resource definition that is installed in the CICS region.

Conditions**INVREQ**

RESP2 value:

- n** The ATTRIBUTES string contains a syntax error, or an error occurred during either the discard or resource definition phase of the processing. The RESP2 value is associated with a message written to the transient data queue CSMT, which identifies more precisely the nature of the error.

LENGERR

RESP2 value:

- 1** The length that you have specified in ATTRLEN is negative.

NOTAUTH

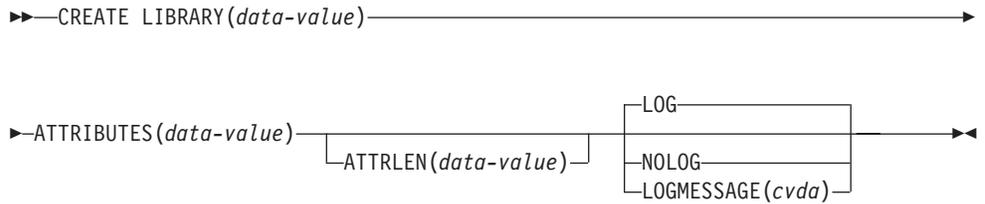
RESP2 values:

- 100** The user associated with the issuing task is not authorized to use this command.
- 101** The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

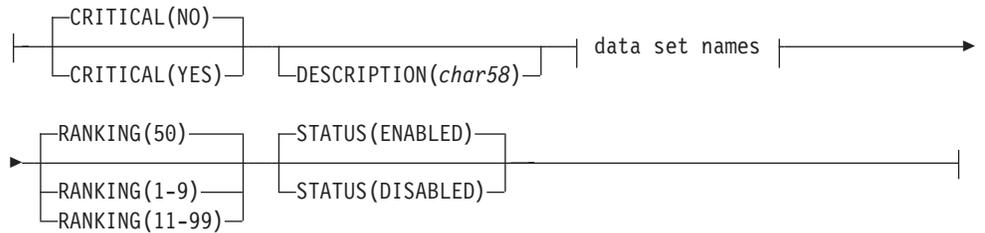
CREATE LIBRARY

Create a LIBRARY resource in the local CICS region.

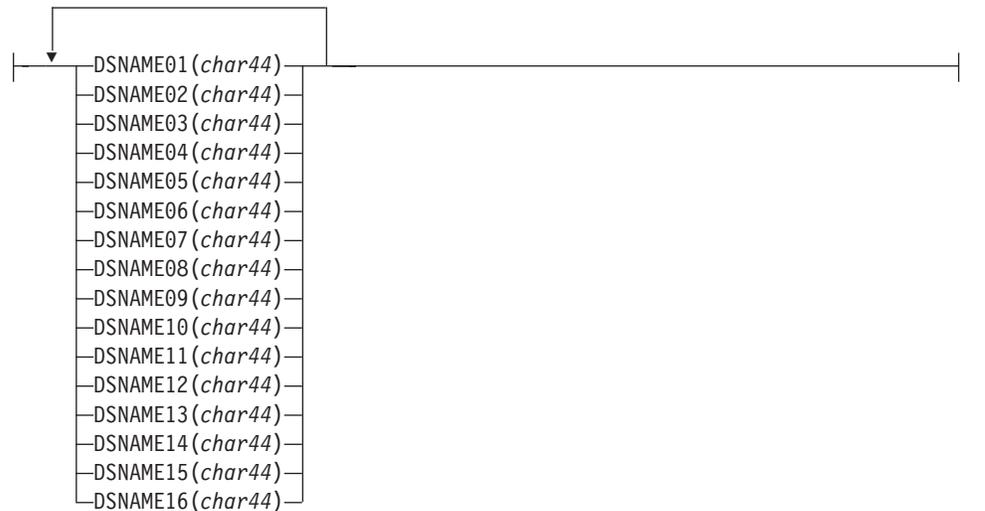
CREATE LIBRARY



CREATE LIBRARY attribute values:



data set names:



Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Description

The `CREATE LIBRARY` command installs a `LIBRARY` resource with the attributes specified on the command. It does not use a resource definition stored in the CSD. `LIBRARY` names must be unique within a CICS region. If the local region already has one of these resources with the name you specify, the new definition replaces the old one, providing the old `LIBRARY` is disabled, otherwise the command is rejected.

Options

ATTRIBUTES (*data-value*)

Specifies the attributes of the LIBRARY resource being added. The list of attributes must be coded as a single character string using the syntax shown in **LIBRARY attributes**. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN (*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length can be from 0 to 32767.

LIBRARY (*data-value*)

Specifies the name of the LIBRARY resource to be added to the CICS region. This name is used as the DD name for the LIBRARY concatenation of data sets, so must conform to the rules for a valid DD name. It is also advisable to avoid using names that will clash with other DD names used by the installation, and the following reserved names are not accepted:

- DFHRPL or any name beginning with 'DFH'
- Any name beginning with 'EYU'
- JOBLIB
- STEPLIB
- JOBCAT
- STEPCAT
- SYSABEND
- SYSUDUMP
- SYSMDUMP
- CEEDUMP
- SYSIN
- SYSOUT
- DUMMY

JOBLIB through to CEEDUMP in this list are DD names reserved by MVS, DFHRPL is reserved for use in the CICS start-up JCL as the name of the static LIBRARY, and SYSIN and SYSOUT and DUMMY are conventionally used as DD cards and therefore inadvisable as LIBRARY names. LIBRARY names beginning with CEE should be used with caution, but are accepted. LIBRARY names beginning with 'DFH' or 'EYU' are not accepted, to ensure that these names remain available in future for use by CICS. The create of the LIBRARY fails if the chosen name conflicts with DD names already in use in the system.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

Note: The audit log messages for LIBRARY resources are written to CSLB regardless of the value of LOGMESSAGE.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier change which affects the LIBRARY search order has not yet completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. Most of the RESP2 values issued by the EXEC CICS CREATE command are associated with a message that is written to transient data queue CSMT. The RESP2 values and the corresponding message numbers are shown in a table in the Appendix C, "RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands," on page 837 topic in the CICS Information Center.

Syntax errors can be caused by the following:

- An invalid LIBRARY name.
- A ranking value is out of the range 1-99.
- A ranking value is the reserved value of 10.
- No DSNAMExx attribute is specified (at least 1 data set name must be provided).
- The LIBRARY name provided is a reserved name

Errors during the discard or resource definition phase can be caused by the following:

- Insufficient storage to create internal control structures for the LIBRARY
- Data set allocation failed
- Data set concatenation failed
- The LIBRARY failed to open
- An existing LIBRARY of the same name already exists and is not disabled
- Write to the CICS catalog failed

- 7 The LOGMESSAGE CVDA value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program was invoked from a remote system by a distributed program link without the SYNCONRETURN option.
- 500 Install failed because an existing LIBRARY of the same name exists, and could not be replaced because it is not disabled.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

CREATE LSRPOOL

Define an LSR pool in the local CICS region.

CREATE LSRPOOL

►► CREATE LSRPOOL(*data-value*)

►► ATTRIBUTES(*data-value*)

- └─ ATTRLEN(*data-value*) ─┘
- └─ LOG ─┘
- └─ NOLOG ─┘
- └─ LOGMESSAGE(*cvda*) ─┘

CREATE LSRPOOL attribute values:

└─ DESCRIPTION(*char58*) ─┘

└─ Data buffer attributes ─┘

└─ Index buffer attributes ─┘

└─ LSRPOOLNUM(1) ─┘

└─ LSRPOOLNUM(1-255) ─┘

└─ MAXKEYLENGTH(0-255) ─┘

└─ SHARELIMIT(50) ─┘

└─ SHARELIMIT(1-100) ─┘

└─ STRINGS(1-255) ─┘

Data buffer attributes:

DATA512(3-32767)
DATA1K(3-32767)
DATA2K(3-32767)
DATA4K(3-32767)
└─ HSDATA4K(0-16777215) ─┘
DATA8K(3-32767)
└─ HSDATA8K(0-16777215) ─┘
DATA12K(3-32767)
└─ HSDATA12K(0-16777215) ─┘
DATA16K(3-32767)
└─ HSDATA16K(0-16777215) ─┘
DATA20K(3-32767)
└─ HSDATA20K(0-16777215) ─┘
DATA24K(3-32767)
└─ HSDATA24K(0-16777215) ─┘
DATA28K(3-32767)
└─ HSDATA28K(0-16777215) ─┘
DATA32K(3-32767)
└─ HSDATA32K(0-16777215) ─┘

Index buffer attributes:

INDEX512(3-32767)	
INDEX1K(3-32767)	
INDEX2K(3-32767)	
INDEX4K(3-32767)	HSINDEX4K(0-16777215)
INDEX8K(3-32767)	HSINDEX8K(0-16777215)
INDEX12K(3-32767)	HSINDEX12K(0-16777215)
INDEX16K(3-32767)	HSINDEX16K(0-16777215)
INDEX20K(3-32767)	HSINDEX20K(0-16777215)
INDEX24K(3-32767)	HSINDEX24K(0-16777215)
INDEX28K(3-32767)	HSINDEX28K(0-16777215)
INDEX32K(3-32767)	HSINDEX32K(0-16777215)

Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use `ATTRIBUTES(data-area)` instead of `ATTRIBUTES(data-value)`.

Description

The **CREATE LSRPOOL** command installs an LSRPOOL definition with the attribute specified on the command and does not use a resource definition stored in the CSD. LSR pools must have unique LSRPOOLNUM values within a CICS region. If the local region already contains a definition with the same LSRPOOLNUM value, the new definition replaces the old one; if not, the new definition is added. Unlike most resource definitions, the name you specify in the LSRPOOL option does not determine replacement; instead the value in LSRPOOLNUM controls the action taken.

Note: When you replace the definition of a pool that is currently open, the new definition does not take effect until the next time the pool is built. The pool is not rebuilt until all of the files that are using it are closed and one is reopened subsequently.

A sync point is implicit in **CREATE LSRPOOL** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not. See "Creating resource definitions" on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES(data-value)

Specifies the attributes of the LSRPOOL that are being added. The list of attributes must be coded as a single character string using the syntax shown in

LSRPOOL attributes. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

Note: You can assign default values for all attributes of an LSRPOOL definition by specifying an ATTRLEN value of 0. You still need to specify the ATTRIBUTES option, however, even though its value is not used.

ATTRLEN (*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length can be from 0 to 32767.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

LSRPOOL (*data-value*)

Specifies the 8-character name of the LSRPOOL definition to be added to the CICS region.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, “RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands,” on page 837 for information about RESP2 values.
- 7 The LOGMESSAGE cvda value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

NOTAUTH

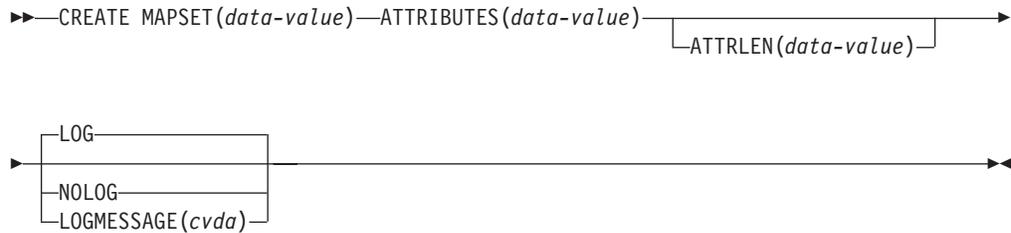
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

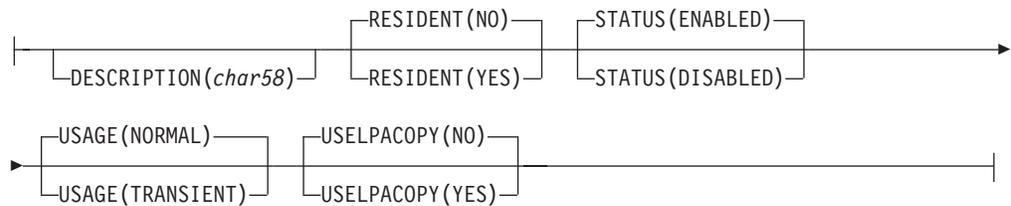
CREATE MAPSET

Define a map set in the local CICS region.

CREATE MAPSET



CREATE MAPSET attribute values:



Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use `ATTRIBUTES(data-area)` instead of `ATTRIBUTES(data-value)`.

Description

The **CREATE MAPSET** command installs a MAPSET definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. Map set names must be unique among map set, program, and partition set names within a CICS region. If the local region already has one of these resources with the name you specify, the new definition replaces the old one; if not, the new definition is added.

A syncpoint is implicit in **CREATE MAPSET** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the `CREATE` executes successfully, and rolled back if not. See "Creating resource definitions" on page 24 for other general rules governing `CREATE` commands.

Options

ATTRIBUTES(data-value)

Specifies the attributes of the MAPSET being added. The list of attributes must be coded as a single character string using the syntax shown in **MAPSET attributes**. See "The `ATTRIBUTES` option" on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

Note: You can assign default values for all attributes of a MAPSET definition by specifying an ATTRLEN value of 0. You still need to specify the ATTRIBUTES option, however, even though its value is not used.

ATTRLEN (*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length can be from 0 to 32 767.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

MAPSET (*data-value*)

specifies the 8-character name of the MAPSET definition to be added to the CICS region.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, "RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands," on page 837 for information on RESP2 values.
- 7 The LOGMESSAGE cvda value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

NOTAUTH

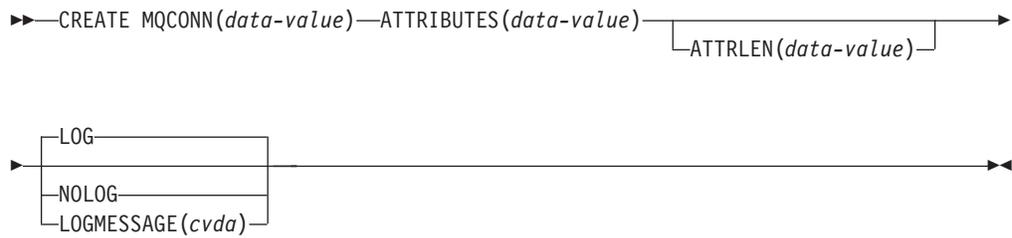
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to create a MAPSET definition with this name.

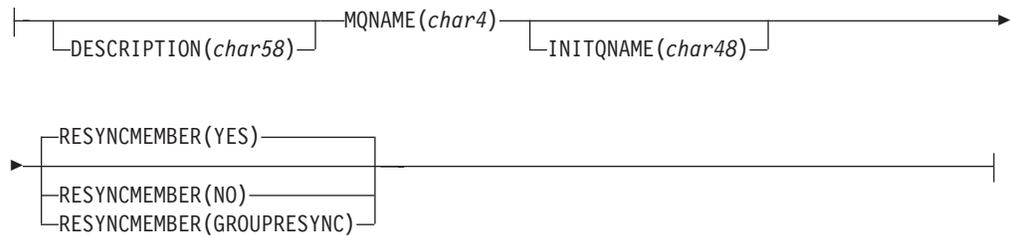
CREATE MQCONN

Define an MQCONN resource in the local CICS region.

CREATE MQCONN



CREATE MQCONN attribute values:



Conditions: INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use `ATTRIBUTES(data-area)` instead of `ATTRIBUTES(data-value)`.

Description

The `CREATE MQCONN` command installs an `MQCONN` resource definition with the attributes specified on the command. It does not use a resource definition stored in the CSD. If an `MQCONN` resource definition is already in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

An `MQCONN` resource definition can be installed only when CICS is not connected to WebSphere® MQ.

A sync point is implicit in `CREATE MQCONN` processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the `CREATE` command is successful, and rolled back if not.

Options

`ATTRIBUTES` (data-value)

Specifies the attributes of the `MQCONN` resource definition being added. You must code the list of attributes as a single character string by using the syntax shown in the syntax diagram.

See the *CICS Resource Definition Guide* for details about specific attributes. You can assign default values for all attributes of an `MQCONN` definition by specifying an `ATTRLEN` value of 0. You must still specify the `ATTRIBUTES` option, however, even though its value is not used.

`ATTRLEN` (data-value)

Specifies the length in bytes of the character string supplied in the `ATTRIBUTES` option, as a halfword binary value. The length can be 0 - 32767.

LOGMESSAGE(*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

MQCONN(*data-value*)

Specifies the 8-character name of the MQCONN resource definition to be added to the CICS region.

Conditions

INVREQ

RESP2 values:

- n** The ATTRIBUTES string contains a syntax error, or an error occurred during either the discard or resource definition phase of the processing.
- 7** The LOGMESSAGE CVDA value is not valid.
- 200** The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET or a program called from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1** The length that you have specified in ATTRLEN is negative.

NOTAUTH

RESP2 values:

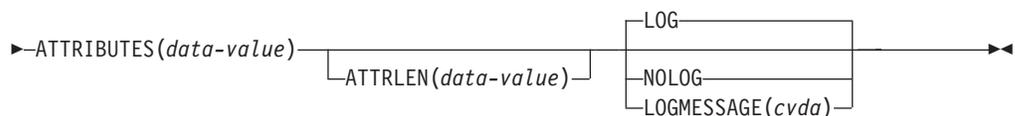
- 100** The user associated with the issuing task is not authorized to use this command.

CREATE PARTITIONSET

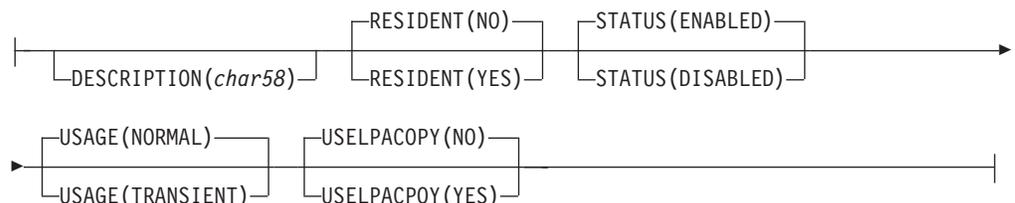
Define a partition set in the local CICS region.

CREATE PARTITIONSET

►► CREATE PARTITIONSET(*data-value*)



CREATE PARTITIONSET attribute values:



Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use `ATTRIBUTES(data-area)` instead of `ATTRIBUTES(data-value)`.

Description

The **CREATE PARTITIONSET** command installs a PARTITIONSET definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. Partition set names must be unique among partition set, map set, and program names within a CICS region. If the local region already has one of these resources with the name you specify, the new definition replaces the old one; if not, the new definition is added.

A syncpoint is implicit in **CREATE PARTITIONSET** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not. See “Creating resource definitions” on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES(data-value)

Specifies the attributes of the PARTITIONSET being added. The list of attributes must be coded as a single character string using the syntax shown in **PARTITIONSET attributes**. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

Note: You can assign default values for all attributes of a PARTITIONSET definition by specifying an ATTRLEN value of 0. You still need to specify the ATTRIBUTES option, however, even though its value is not used.

ATTRLEN(data-value)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length can be from 0 to 32 767.

LOGMESSAGE(cvda)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

PARTITIONSET(data-value)

Specifies the 8-character name of the PARTITIONSET definition to be added to the CICS region.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n** There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, "RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands," on page 837 for information on RESP2 values.
- 7** The LOGMESSAGE cvda value is not valid.
- 200** The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1** The length you have specified in ATTRLEN is negative.

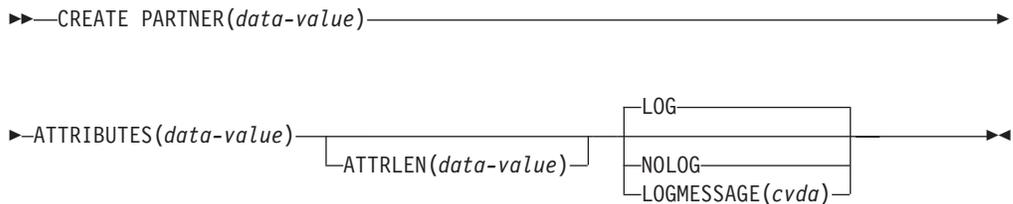
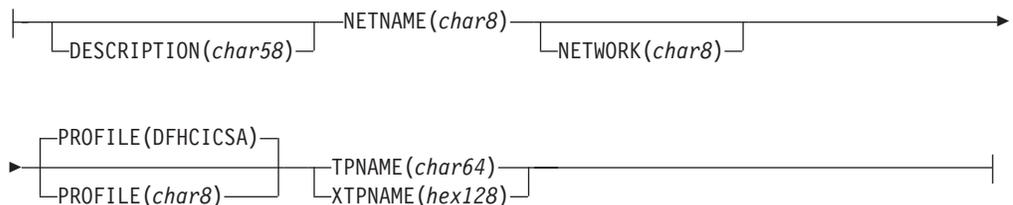
NOTAUTH

RESP2 values:

- 100** The user associated with the issuing task is not authorized to use this command.
- 101** The user associated with the issuing task is not authorized to create a PARTITIONSET definition with this name.

CREATE PARTNER

Define a PARTNER in the local CICS region.

CREATE PARTNER**CREATE PARTNER attribute values:****Conditions:** ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use **ATTRIBUTES(data-area)** instead of **ATTRIBUTES(data-value)**.

Description

The **CREATE PARTNER** command installs a PARTNER definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is already a partner with the name you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

A syncpoint is implicit in **CREATE PARTNER** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not. See “Creating resource definitions” on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES (*data-value*)

Specifies the attributes of the PARTNER being added. The list of attributes must be coded as a single character string using the syntax shown in **PARTNER attributes**. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN (*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length must not exceed 32 767 bytes.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

PARTNER (*data-value*)

Specifies the 8-character name of the PARTNER definition to be added to the CICS region.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, “RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands,” on page 837 for information on RESP2 values.
- 7 The LOGMESSAGE cvda value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

NOTAUTH

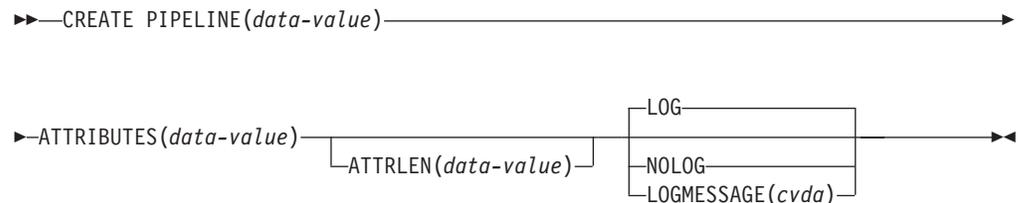
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

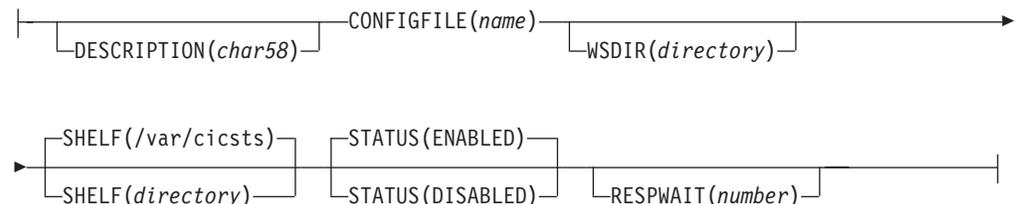
CREATE PIPELINE

Define a PIPELINE in the local CICS region.

CREATE PIPELINE



CREATE PIPELINE attribute values:



Conditions: INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use `ATTRIBUTES(data-area)` instead of `ATTRIBUTES(data-value)`.

Description

The **CREATE PIPELINE** command installs a PIPELINE definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is already a PIPELINE with the name you specify in the local CICS region, and the existing PIPELINE is disabled, the new definition replaces the old one; if an existing PIPELINE is not disabled, the CREATE command fails.

A syncpoint is implicit in **CREATE PIPELINE** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not. See "Creating resource definitions" on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES (*data-value*)

Specifies the attributes of the PIPELINE being added. The list of attributes must be coded as a single character string using the syntax shown in **PIPELINE attributes**. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN (*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length must not exceed 32 767 bytes.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

PIPELINE (*data-value*)

Specifies the 8-character name of the PIPELINE definition to be added to the CICS region.

Conditions

INVREQ

RESP2 values:

n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. The RESP2 value is associated with a message written to the transient data queue CSMT which identifies more precisely the nature of the error. See Appendix C, “RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands,” on page 837 for information on RESP2 values.

200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

612 Installation of this PIPELINE failed because it already exists

LENGERR

RESP2 values:

1 The length you have specified in ATTRLEN is negative.

NOTAUTH

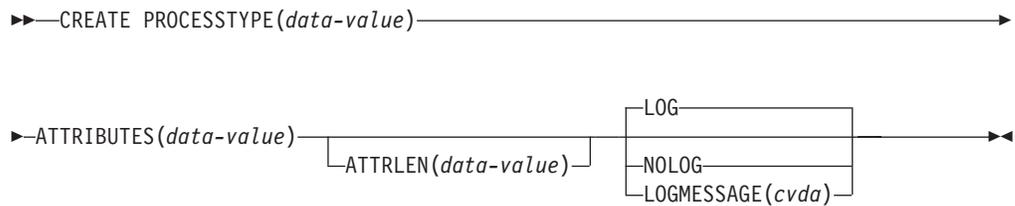
RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

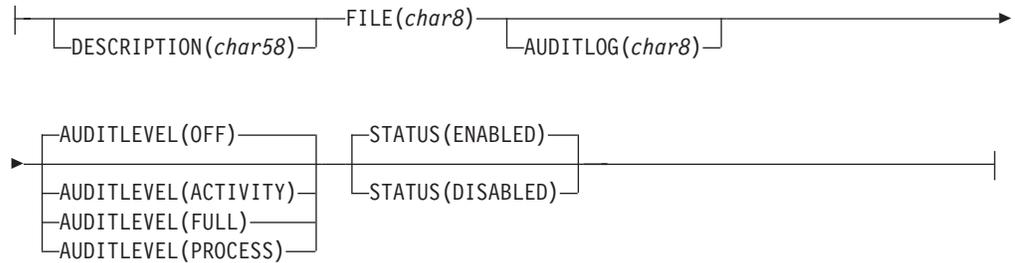
CREATE PROCESSTYPE

Define a PROCESSTYPE in the local CICS region.

CREATE PROCESSTYPE



CREATE PROCESSTYPE attribute values:



Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use **ATTRIBUTES**(*data-area*) instead of **ATTRIBUTES**(*data-value*).

Description

The CREATE PROCESSTYPE command installs a PROCESSTYPE definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is already a process-type with the name you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

A syncpoint is implicit in CREATE PROCESSTYPE processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not. See “Creating resource definitions” on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES(*data-value*)

Specifies the attributes of the PROCESSTYPE being added. The list of attributes must be coded as a single character string using the syntax shown in **PROCESSTYPE attributes**. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN(*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length may not exceed 32 767 bytes.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

PROCESSTYPE (*data-value*)

Specifies the 1- to 8-character name of the PROCESSTYPE definition to be added to the CICS region. The acceptable characters are A-Z a-z 0-9 \$ @ # . / - _ % & ? ! : | " = ~ , ; < >. Leading and embedded blank characters are not permitted. If the name supplied is less than eight characters, it is padded with trailing blanks up to eight characters.

Conditions**ILLOGIC**

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, "RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands," on page 837 for information on RESP2 values.
- 7 The LOGMESSAGE cvda value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

NOTAUTH

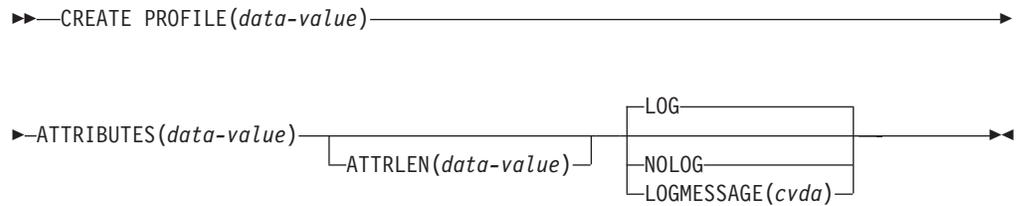
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to create a PROCESSTYPE definition with this name.
- 102 The caller does not have surrogate authority to install the resource with the particular userid.

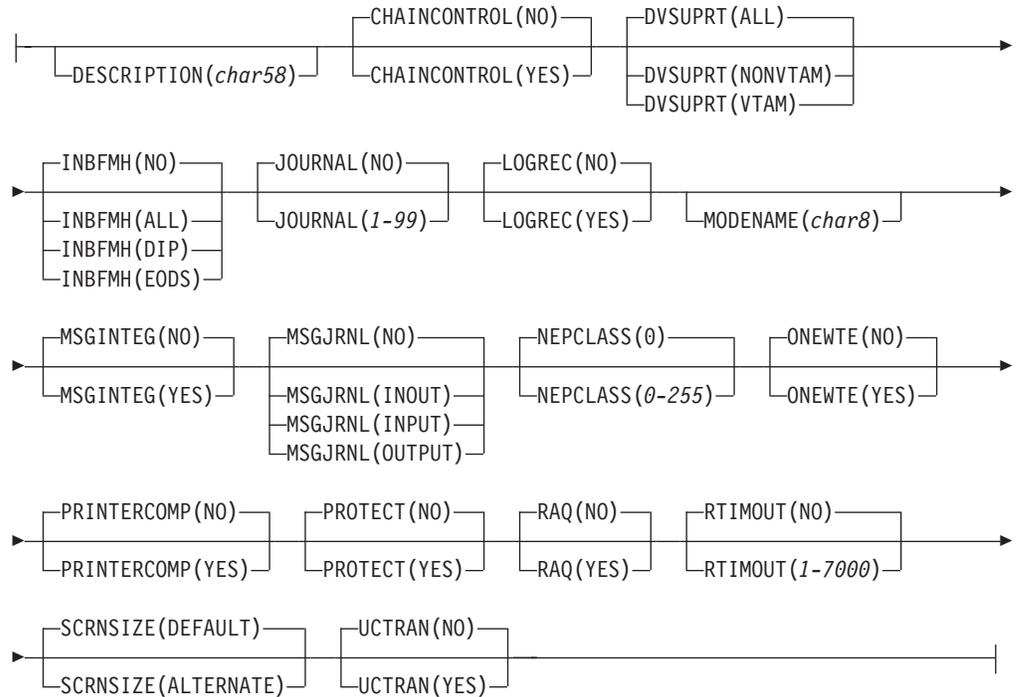
CREATE PROFILE

Define a transaction PROFILE in the local CICS region.

CREATE PROFILE



CREATE PROFILE attribute values:



Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use **ATTRIBUTES(data-area)** instead of **ATTRIBUTES(data-value)**.

Note: VTAM is now z/OS Communications Server.

Description

The CREATE PROFILE command installs a PROFILE definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is already a profile with the name you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

A syncpoint is implicit in CREATE PROFILE processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not. See "Creating resource definitions" on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES (*data-value*)

Specifies the attributes of the PROFILE being added. The list of attributes must be coded as a single character string using the syntax shown in **PROFILE attributes**. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

Note: You can assign default values for all attributes of a PROFILE definition by specifying an ATTRLEN value of 0. You still need to specify the ATTRIBUTES option, however, even though its value is not used.

ATTRLEN (*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length can be from 0 to 32 767.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

PROFILE (*data-value*)

Specifies the 8-character name of the PROFILE definition to be added to the CICS region.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, “RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands,” on page 837 for information on RESP2 values.
- 7 The LOGMESSAGE cvda value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

NOTAUTH

RESP2 values:

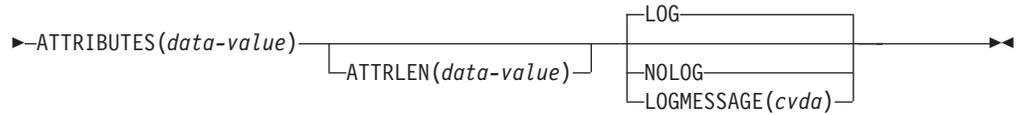
- 100 The user associated with the issuing task is not authorized to use this command.

CREATE PROGRAM

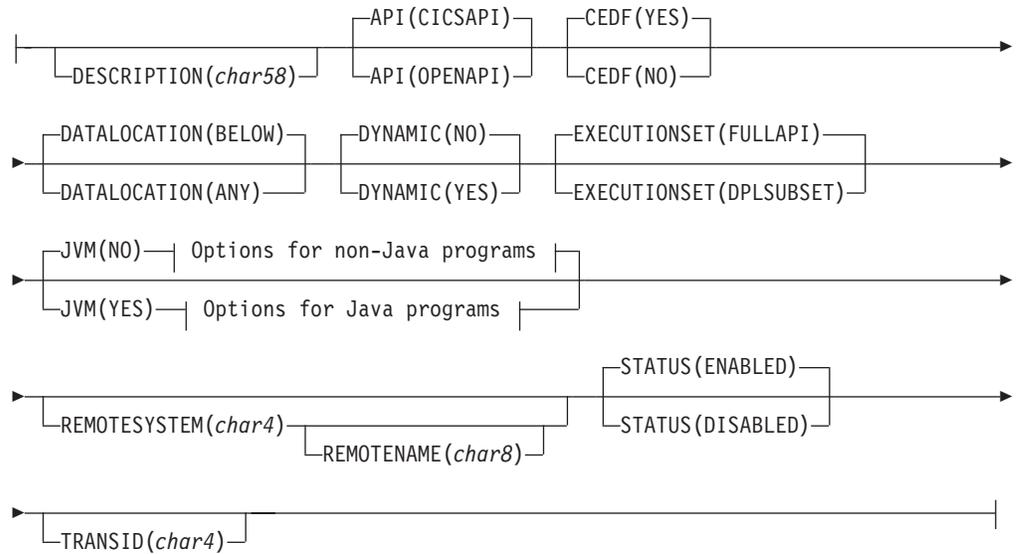
Define a PROGRAM in the local CICS region.

CREATE PROGRAM

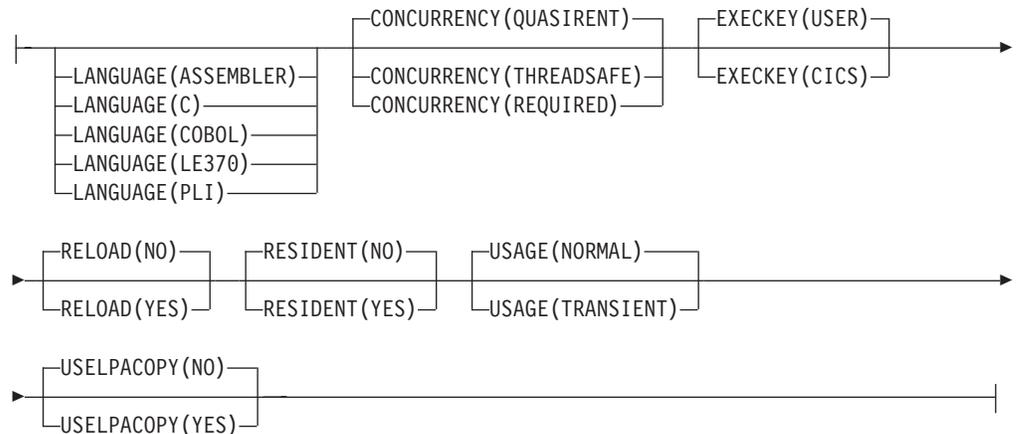
►► CREATE PROGRAM(*data-value*)



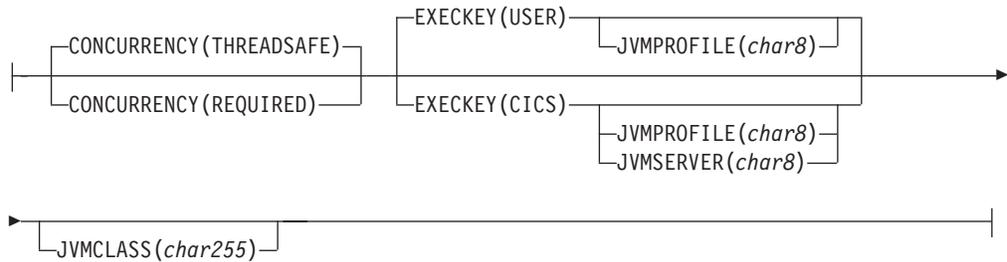
CREATE PROGRAM attribute values:



Attributes for non-Java programs:



Options for Java programs:



Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use **ATTRIBUTES**(*data-area*) instead of **ATTRIBUTES**(*data-value*).

Description

The **CREATE PROGRAM** command installs a PROGRAM definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. Program names must be unique among program, map set, and partition set names within a CICS region. If the local region already has one of these resources with the name you specify, the new definition replaces the old one; if not, the new definition is added.

A sync point is implicit in **CREATE PROGRAM** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not. See “Creating resource definitions” on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES(*data-value*)

Specifies the attributes of the PROGRAM being added. The list of attributes must be coded as a single character string using the syntax shown in **PROGRAM attributes**. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and PROGRAM attributes in the Resource Definition Guide for details about specific attributes.

Note: You can assign default values for all attributes of a PROGRAM definition by specifying an ATTRLEN value of 0. You must still specify the ATTRIBUTES option, however, even though its value is not used.

ATTRLEN(*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length can be from 0 to 32767.

LOGMESSAGE(*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

PROGRAM(*data-value*)

Specifies the 8-character name of the PROGRAM definition to be added to the CICS region.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, "RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands," on page 837 for information about RESP2 values.
- 7 The LOGMESSAGE cvda value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.
- 407 The command specified a value for both the JVMSERVER and JVMPROFILE options. You cannot set both values for a Java program.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

NOTAUTH

RESP2 values:

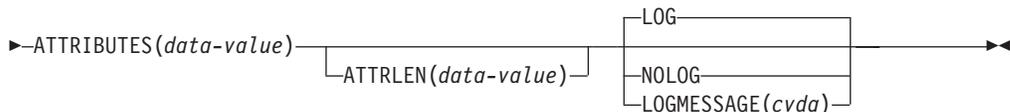
- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to create a PROGRAM definition with this name.

CREATE REQUESTMODEL

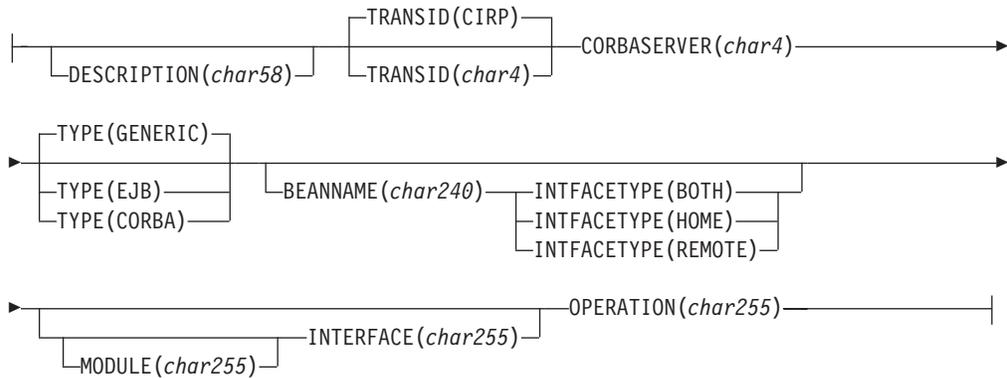
Define a request model in the local CICS region.

CREATE REQUESTMODEL

▶▶—CREATE REQUESTMODEL(*data-value*)—————▶▶



CREATE REQUESTMODEL attribute values:



Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use **ATTRIBUTES**(*data-area*) instead of **ATTRIBUTES**(*data-value*).

Description

The CREATE REQUESTMODEL command installs a REQUESTMODEL definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is already a request model with the name you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

A syncpoint is implicit in CREATE REQUESTMODEL processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not. See “Creating resource definitions” on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES(*data-value*)

Specifies the attributes of the REQUESTMODEL being added. The list of attributes must be coded as a single character string using the syntax shown in **REQUESTMODEL attributes**. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN(*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length can be from 0 to 32767.

LOGMESSAGE(*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

REQUESTMODEL (*data-value*)

Specifies the 8-character name of the REQUESTMODEL definition to be added to the CICS region.

Conditions**ILLOGIC**

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, "RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands," on page 837 for information on RESP2 values.
- 7 The LOGMESSAGE CVDA value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.
- 611 Installation of the REQUESTMODEL definition *r1* failed because a duplicate pattern already exists in *r2*.
- 618 Installation of the REQUESTMODEL definition *r1* failed because it is not a valid REQUESTMODEL for this level of CICS.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

NOTAUTH

RESP2 values:

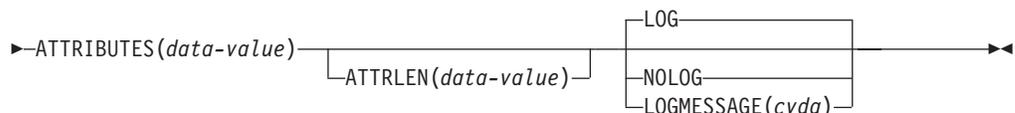
- 100 The user associated with the issuing task is not authorized to use this command.

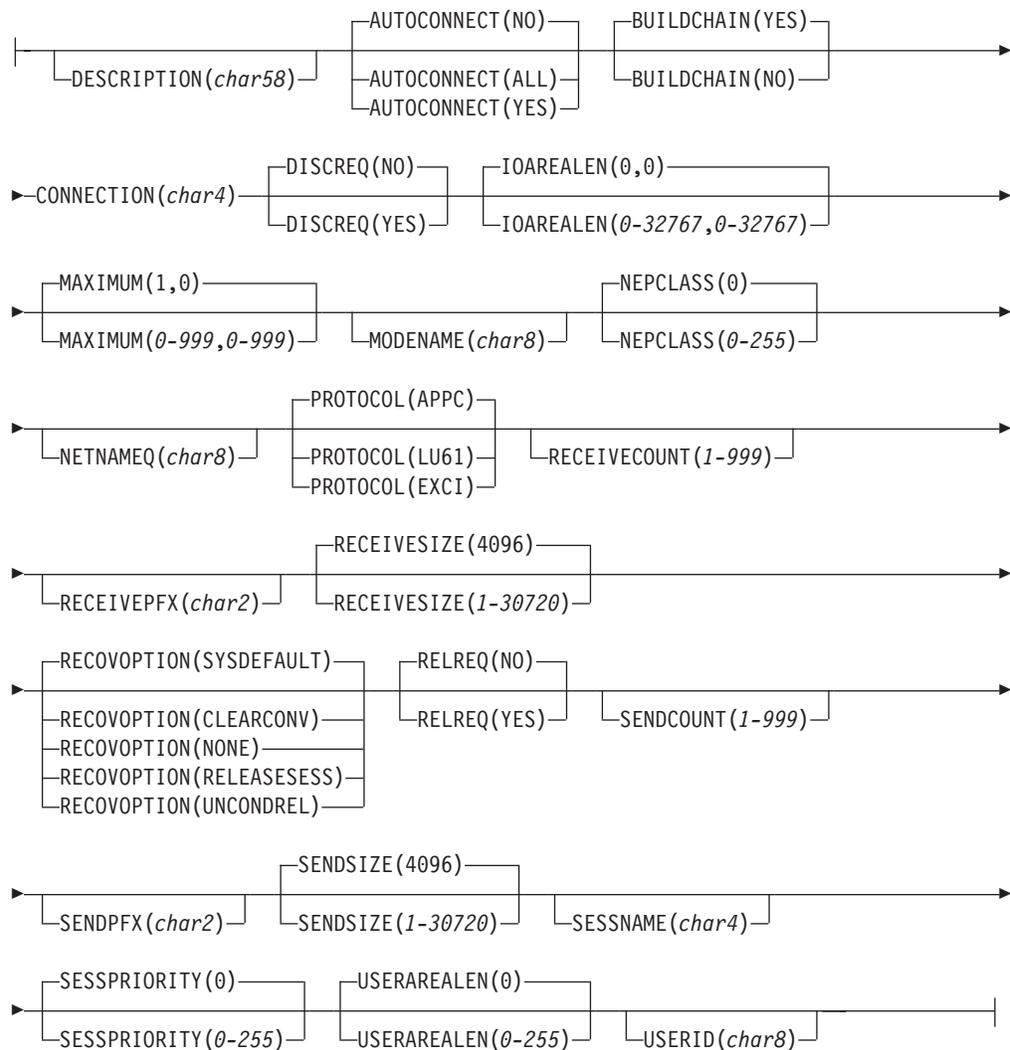
CREATE SESSIONS

Add a session group to the CONNECTION definition being created.

CREATE SESSIONS

►►—CREATE SESSIONS—(*—data-value—*)—————►

**CREATE SESSIONS attribute values:**



Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use **ATTRIBUTES**(*data-area*) instead of **ATTRIBUTES**(*data-value*).

Description

The **CREATE SESSIONS** command installs a **SESSIONS** definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. You can use it only after issuing the initial **CREATE CONNECTION** command that defines the attributes of a connection and before the final **CREATE CONNECTION COMPLETE** (or **DISCARD**) command that ends the process.

The sessions you define always belong to the current connection, and the name that you specify in the **CONNECTION** option within your **ATTRIBUTES** string must match the name of the connection specified in the preceding **CREATE CONNECTION** command. See “**CREATE CONNECTION**” on page 45 for rules about the order of the commands that build a connection, and “**Creating resource definitions**” on page 24 for general rules governing **CREATE** commands.

Options

ATTRIBUTES (*data-value*)

specifies the attributes of the group of SESSIONS being added. The list of attributes must be coded as a single character string using the syntax shown in **SESSIONS attributes**. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN (*data-value*)

specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length may not exceed 32767 bytes.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

SESSIONS (*data-value*)

specifies the 8-character name of the SESSIONS definition to be added to CONNECTION definition under construction. The name of a sessions group needs to be unique only within the current CONNECTION definition, and the group is always added unless you repeat a session name within a connection. In this case, the last successful SESSIONS definition of the same name is the one that is used.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because no CREATE CONNECTION ATTRIBUTES command has been issued, or the CONNECTION name specified in the ATTRIBUTES argument of this command does not match the name of the connection assigned in the CREATE CONNECTION command.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, “RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands,” on page 837 for information on RESP2 values.
- 7 The LOGMESSAGE cvda value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

NOTAUTH

RESP2 values:

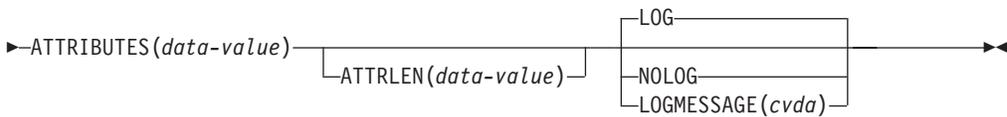
- 100 The user associated with the issuing task is not authorized to use this command.
- 102 The user of the transaction issuing the CREATE SESSIONS command is not an authorized surrogate of the user specified in USERID.

CREATE TCPIP SERVICE

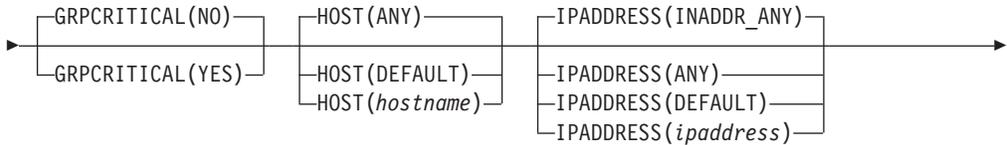
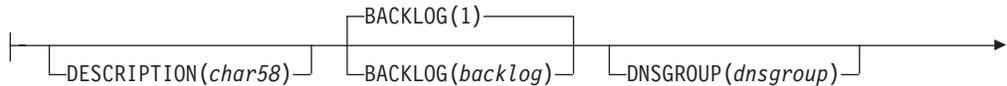
Define a TCP/IP service in the local CICS region.

CREATE TCPIP SERVICE

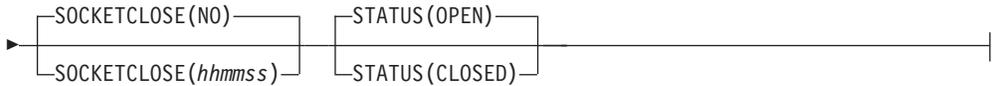
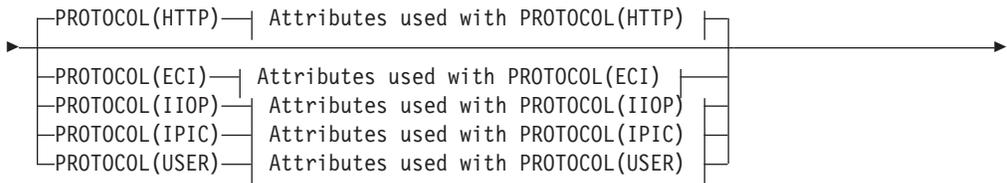
►► CREATE TCPIP SERVICE (*data-value*)



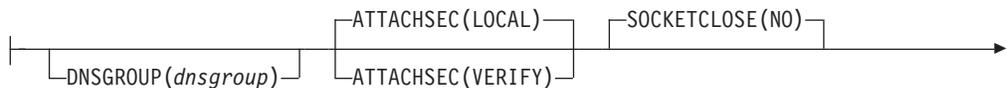
CREATE TCPIP SERVICE attribute values:



►► PORTNUMBER (*port*)

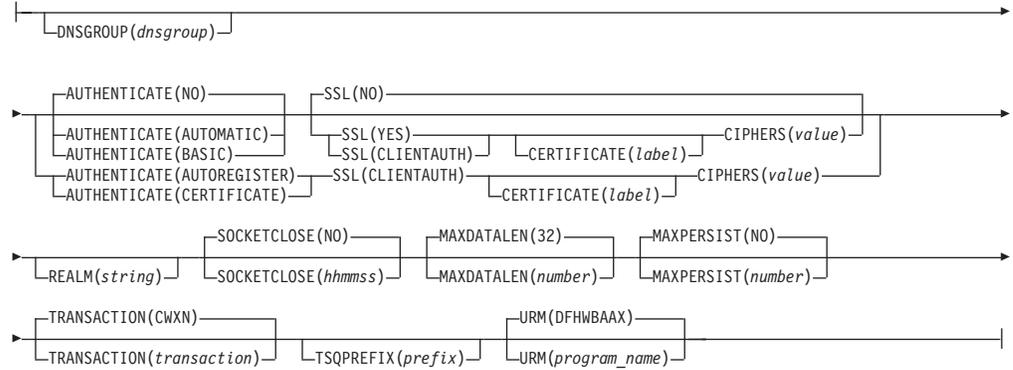


Attributes used with PROTOCOL (ECI):

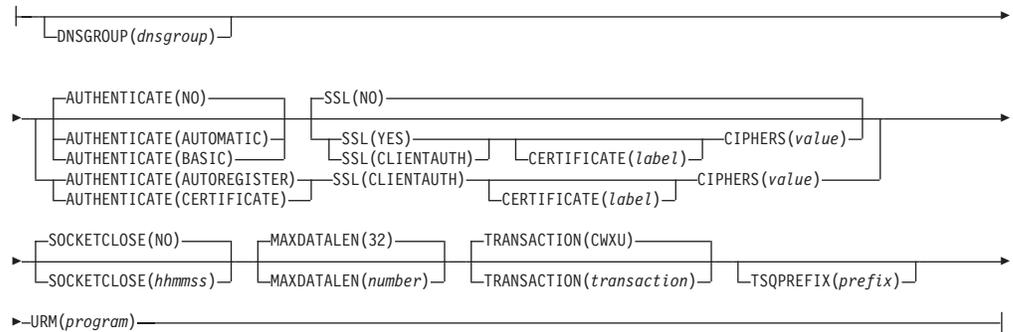




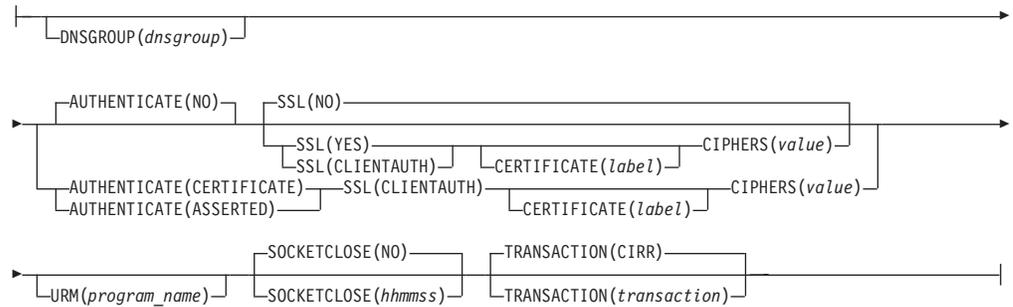
Attributes used with PROTOCOL(HTTP):



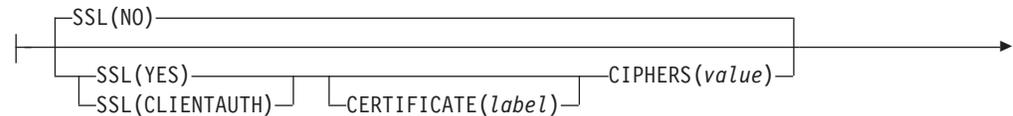
Attributes used with PROTOCOL(USER):

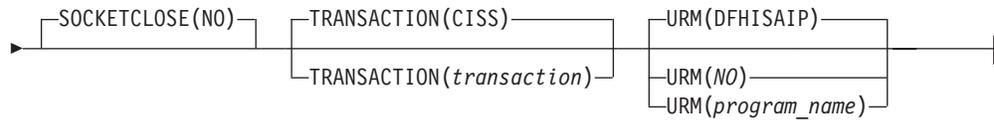


Attributes used with PROTOCOL(IIOP):



Attributes used with PROTOCOL(IPIC):





Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use **ATTRIBUTES**(*data-area*) instead of **ATTRIBUTES**(*data-value*).

Description

The **CREATE TCPIP SERVICE** command installs a TCPIP SERVICE definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If a TCP/IP service already exists with the name that you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

A sync point is implicit in **CREATE TCPIP SERVICE** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE ran successfully, and rolled back if not.

See “Creating resource definitions” on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES(*data-value*)

Specifies the attributes of the TCPIP SERVICE being added. The list of attributes must be coded as a single character string. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN(*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length must not exceed 32,767 bytes.

LOGMESSAGE(*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

TCPIP SERVICE(*data-value*)

Specifies the 8-character name of the TCPIP SERVICE definition to be added to the CICS region.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot run because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n** The ATTRIBUTES string contains a syntax error, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, "RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands," on page 837 for information on RESP2 values.
- 7** The LOGMESSAGE CVDA value is not valid.
- 200** The command ran in a program defined with an EXECUTIONSET value of DPLSUBSET or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1** The length that you have specified in ATTRLEN is negative.

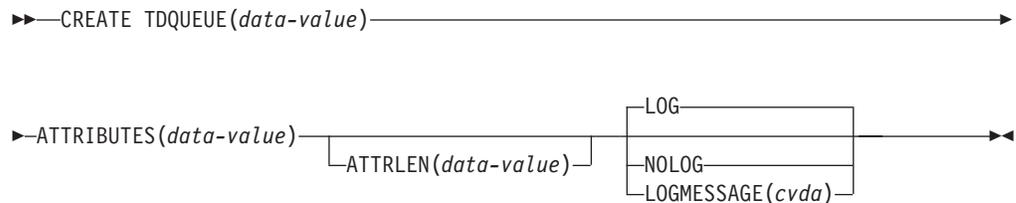
NOTAUTH

RESP2 values:

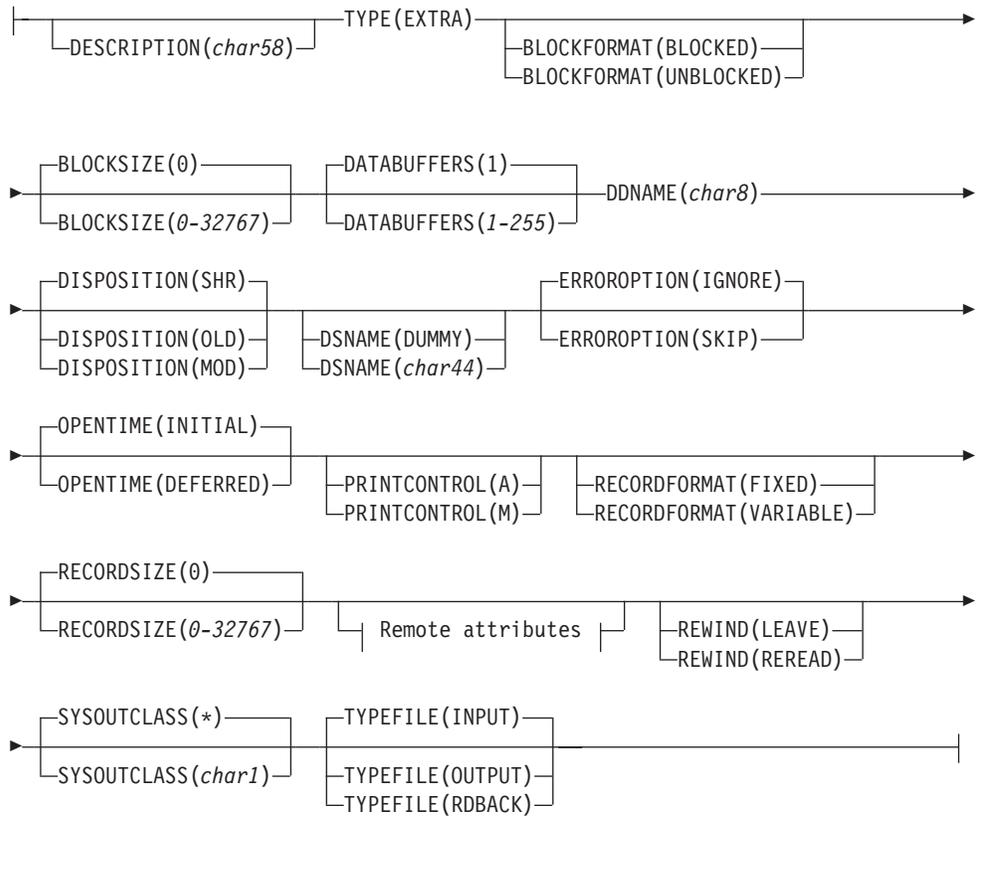
- 100** The user associated with the issuing task is not authorized to use this command.

CREATE TDQUEUE

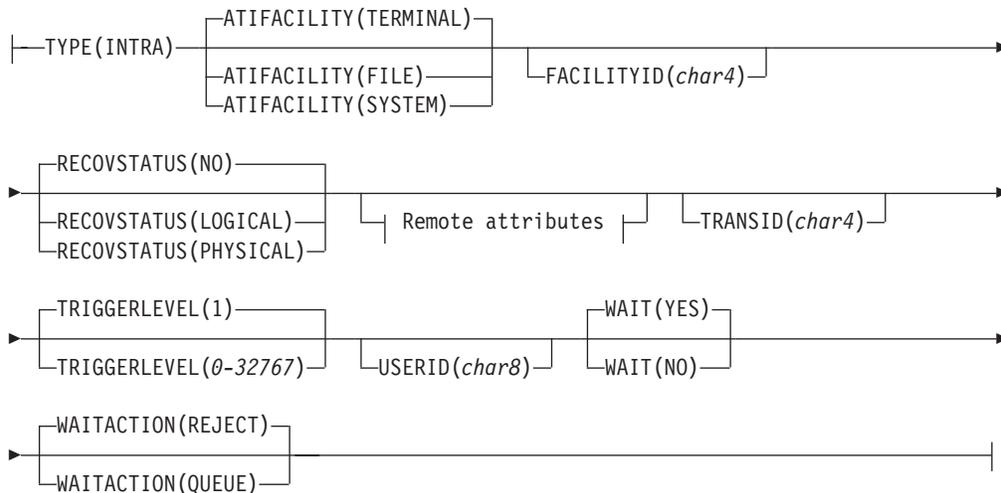
Define a transient data queue in the local CICS region.

CREATE TDQUEUE**Conditions:** ILLOGIC, INVREQ, LENGERR, NOTAUTH

CREATE TDQUEUE attribute values for extra-partition queues:



CREATE TDQUEUE attribute values for intra-partition queues:



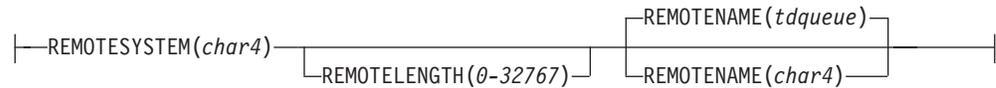
CREATE TDQUEUE attribute values for indirect queues:



CREATE TDQUEUE attribute values for remote queues of unspecified TYPE:



Remote attributes:



Note to COBOL programmers: In the syntax above, you must use `ATTRIBUTES(data-area)` instead of `ATTRIBUTES(data-value)`.

Description

The `CREATE TDQUEUE` command installs a `TDQUEUE` definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is already a transient data queue with the name you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

A syncpoint is implicit in `CREATE TDQUEUE` processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the `CREATE` executes successfully, and rolled back if not. See “Creating resource definitions” on page 24 for other general rules governing `CREATE` commands.

Options

`ATTRIBUTES(data-value)`

specifies the attributes of the queue being added. The list of attributes must be coded as a single character string and must include the `TYPE` option unless the queue is remote. The remaining attributes depend on the queue type; use the syntax shown in the figure (**extra-partition**, **intra-partition**, or **indirect**) that corresponds to your `TYPE` value. If the queue is remote, you still can specify `TYPE` and use the appropriate syntax, but you can also use the briefer form labelled **remote queues of unspecified TYPE**. See “The `ATTRIBUTES` option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

`ATTRLEN(data-value)`

specifies the length in bytes of the character string supplied in the `ATTRIBUTES` option, as a halfword binary value. The length may not exceed 32767 bytes.

`LOGMESSAGE(cvda)`

Specifies whether CICS logs the attributes used for the resource that is created. `CVDA` values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

TDQUEUE (*data-value*)

specifies the 4-character name of the TDQUEUE definition to be added to the CICS region.

Conditions**ILLOGIC**

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, "RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands," on page 837 for information on RESP2 values.
- 7 The LOGMESSAGE cvda value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

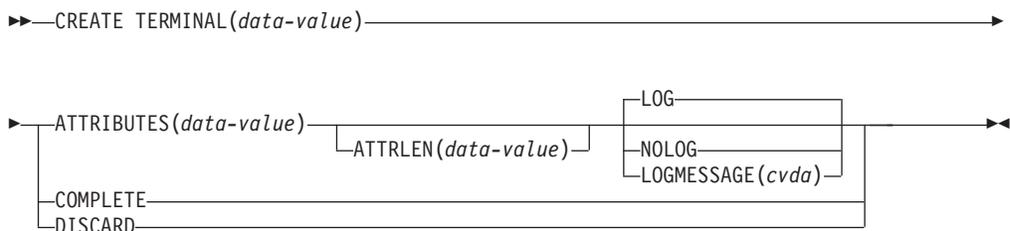
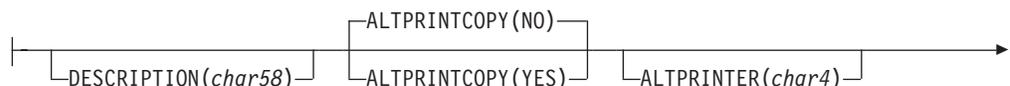
NOTAUTH

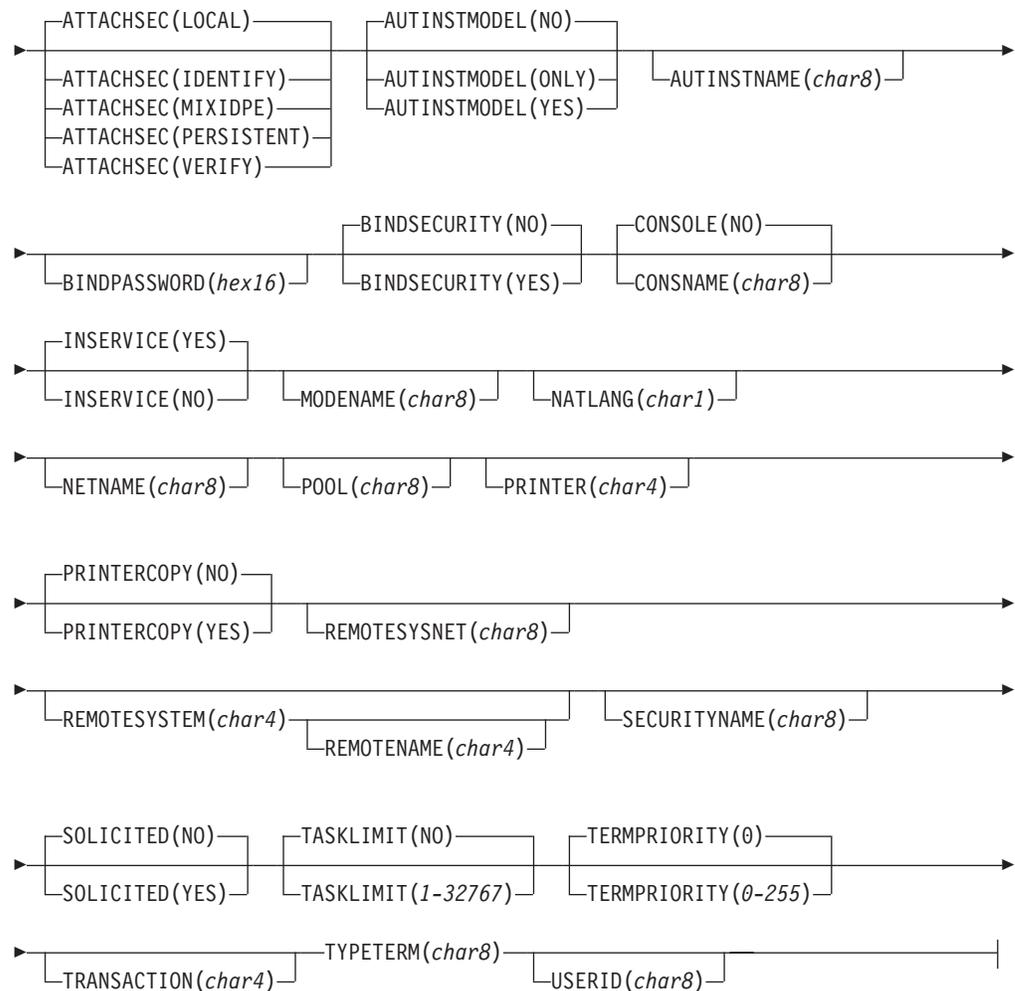
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to create a TDQUEUE definition with this name.
- 102 The user of the transaction issuing the CREATE TDQUEUE command is not an authorized surrogate of the user specified in USERID.

CREATE TERMINAL

Define a TERMINAL in the local CICS region.

CREATE TERMINAL**CREATE TERMINAL attribute values:**



Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use **ATTRIBUTES**(*data-area*) instead of **ATTRIBUTES**(*data-value*).

Description

The CREATE TERMINAL command installs a TERMINAL definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. You can use them either to define individual terminals or a pool of terminals.

The POOL attribute determines which mode you are using. Without it, each command defines a single, independent terminal. If there is already a terminal with the name you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

To define a pool, you issue one CREATE TERMINAL ATTRIBUTES command for each terminal in the pool, specifying the same POOL value in the ATTRIBUTES string. After all of the terminals are defined, you issue CREATE TERMINAL COMPLETE; CICS collects but does not install the TERMINAL definitions until the COMPLETE command. At this point, if there was a pool of the same name in the

local CICS region, CICS deletes all of its terminals and installs the new definitions; if not, it adds the new definitions. Consequently, pool terminals must be defined all at once; you cannot add terminals to an existing pool or include a terminal with the same name as an existing non-pool terminal.

During the time the pool is being built, you must not:

- Change or omit the pool name
- Define other resources of any type, including terminals outside the current pool
- Issue a SYNCPOINT (or any command that implies one)
- Terminate your task (normally)

However, if you encounter an error or problem during the course of building a pool, you can terminate the process at any point by issuing a CREATE TERMINAL DISCARD command. If you do this, CICS discards the partial pool definition, including all of its terminals.

A syncpoint is implicit in CREATE TERMINAL processing, as in other CREATE commands, except when an exception condition is detected early in the processing. Uncommitted changes to recoverable resources are committed when definitions are processed successfully, and rolled back if not or if you specify DISCARD. For non-pool terminals, the syncpoint occurs on each CREATE command. When you are building a pool, however, it occurs only on the command that ends the pool definition, whether you specify COMPLETE or DISCARD. See “Creating resource definitions” on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES(*data-value*)

specifies the attributes of the TERMINAL being added. The list of attributes must be coded as a single character string using the syntax shown in **TERMINAL attributes**. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN(*data-value*)

specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length may not exceed 32767 bytes.

COMPLETE

specifies that the terminal pool definition under construction is complete. It can be used only after the last terminal of a pool has been defined.

DISCARD

specifies that the terminal pool definition under construction is not to be completed, and all of the TERMINAL definitions issued since the pool was started are to be discarded and *not* added.

LOGMESSAGE(*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

TERMINAL(*data-value*)

specifies the 4-character name of the TERMINAL definition to be added.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, "RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands," on page 837 for information on RESP2 values.
- 7 The LOGMESSAGE cvda value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length specified in ATTRLEN is negative.

NOTAUTH

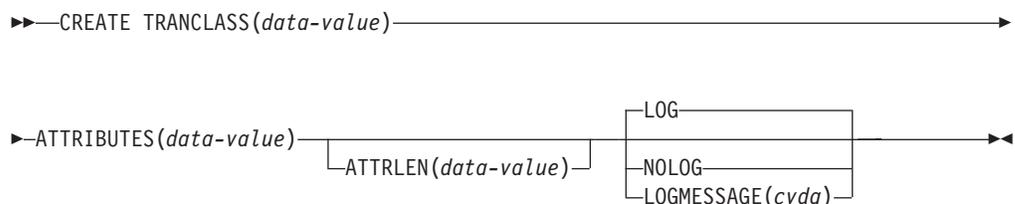
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 102 The user associated with the task issuing the CREATE TERMINAL command is not an authorized surrogate of the user specified in USERID.

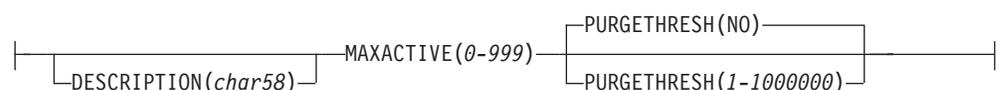
CREATE TRANCLASS

Define a transaction class in the local CICS region.

CREATE TRANCLASS



CREATE TRANCLASS attribute values:



Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use `ATTRIBUTES(data-area)` instead of `ATTRIBUTES(data-value)`.

Description

The `CREATE TRANCLASS` command installs a `TRANCLASS` definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is already a transaction class with the name you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

A syncpoint is implicit in `CREATE TRANCLASS` processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the `CREATE` executes successfully, and rolled back if not. See “Creating resource definitions” on page 24 for other general rules governing `CREATE` commands.

Options

`ATTRIBUTES(data-value)`

specifies the attributes of the `TRANCLASS` being added. The list of attributes must be coded as a single character string using the syntax shown in **TRANCLASS attributes**. See “The `ATTRIBUTES` option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

`ATTRLEN(data-value)`

specifies the length in bytes of the character string supplied in the `ATTRIBUTES` option, as a halfword binary value. The length may not exceed 32767 bytes.

`LOGMESSAGE(cvda)`

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

`TRANCLASS(data-value)`

specifies the 8-character name of the `TRANCLASS` definition to be added to the CICS region.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier `CONNECTION` or `TERMINAL` pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the `ATTRIBUTES` string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, “RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands,” on page 837 for information on RESP2 values.
- 7 The `LOGMESSAGE cvda` value is not valid.

- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

NOTAUTH

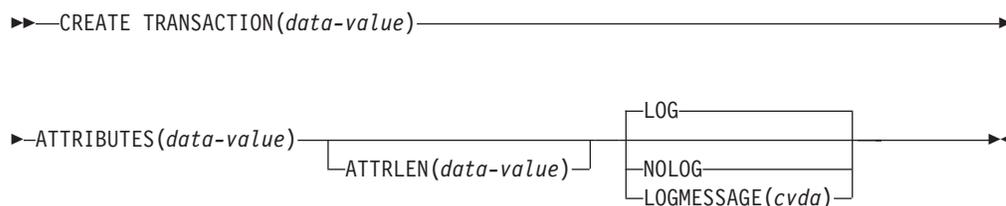
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

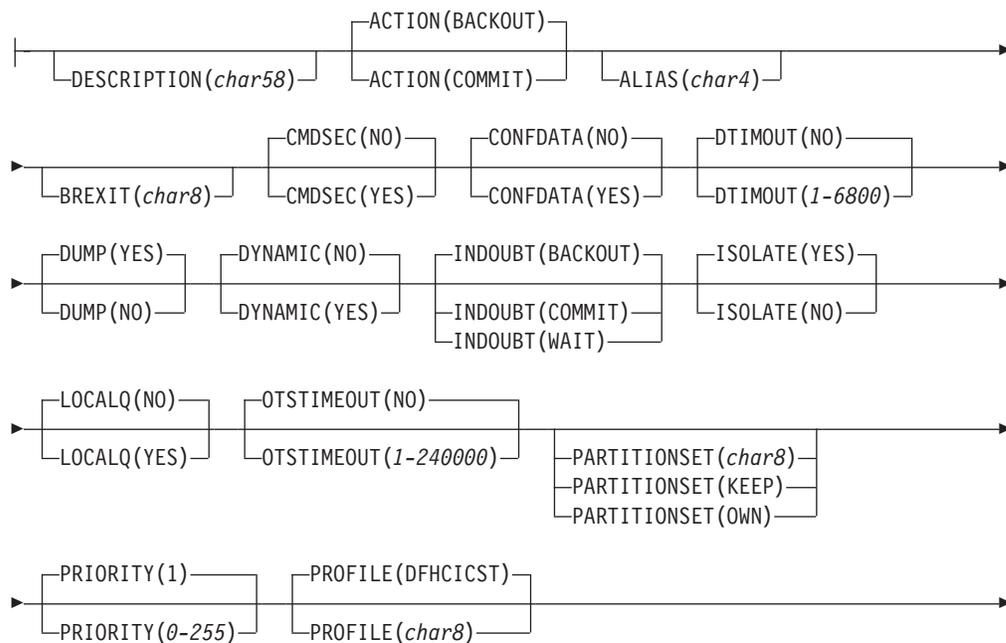
CREATE TRANSACTION

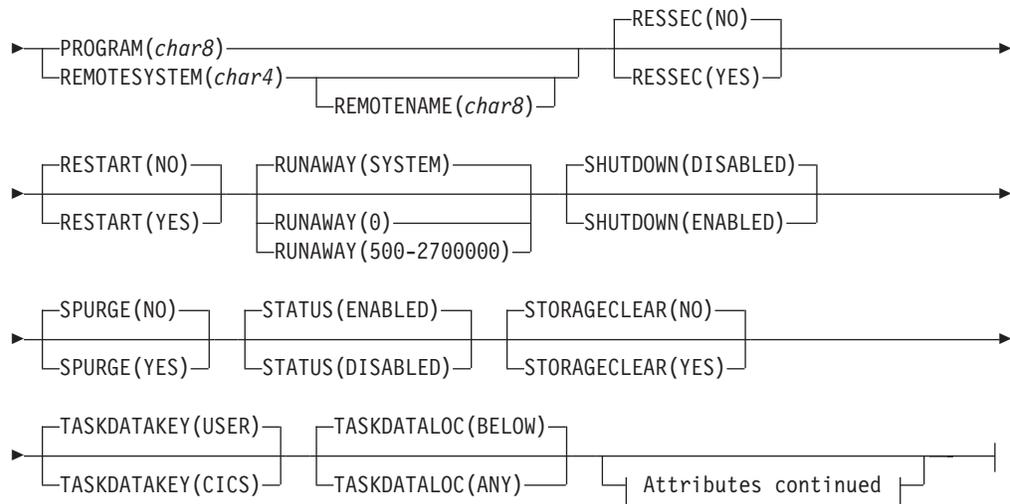
Define a TRANSACTION in the local CICS region.

CREATE TRANSACTION

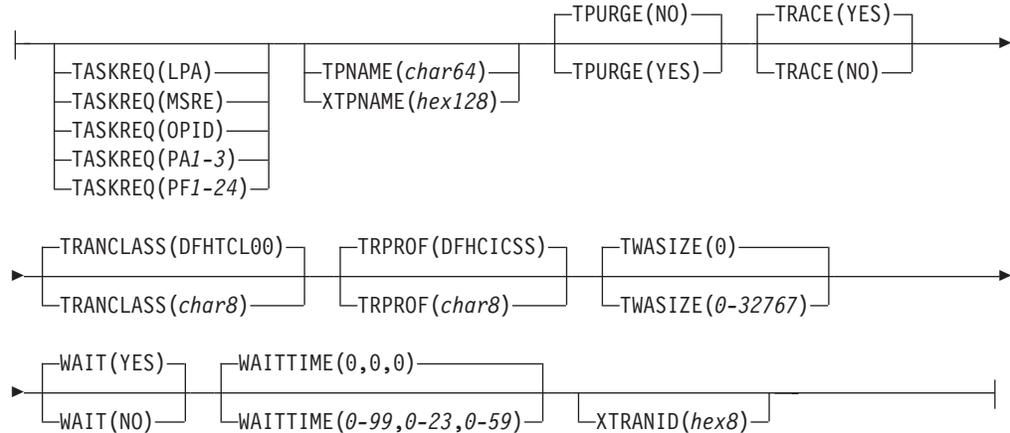


CREATE TRANSACTION attribute values:





Attributes continued:



Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use **ATTRIBUTES**(*data-area*) instead of **ATTRIBUTES**(*data-value*).

Description

The CREATE TRANSACTION command installs a TRANSACTION definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is no transaction with the name you specify in the local CICS region, the new definition is added. If there is, the new definition replaces the old one. However, it does not apply to tasks already in flight, which continue to use the definition under which they were initiated.

A syncpoint is implicit in CREATE TRANSACTION processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not. See “Creating resource definitions” on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES (*data-value*)

specifies the attributes of the TRANSACTION being added. The list of attributes must be coded as a single character string using the syntax shown in **TRANSACTION attributes**. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN (*data-value*)

specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length may not exceed 32767 bytes.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

TRANSACTION (*data-value*)

specifies the 4-character name of the TRANSACTION definition to be added to the CICS region.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, “RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands,” on page 837 for information on RESP2 values.
- 7 The LOGMESSAGE cvda value is not valid.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

NOTAUTH

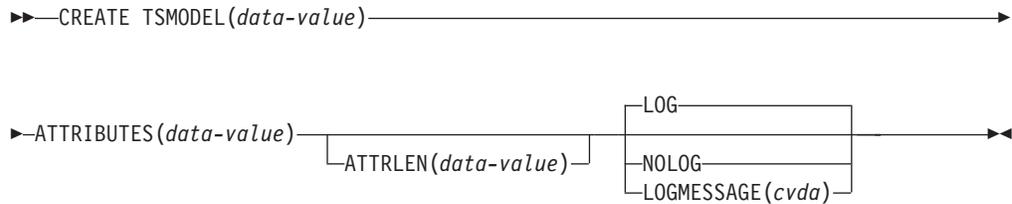
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to create a TRANSACTION definition with this name.

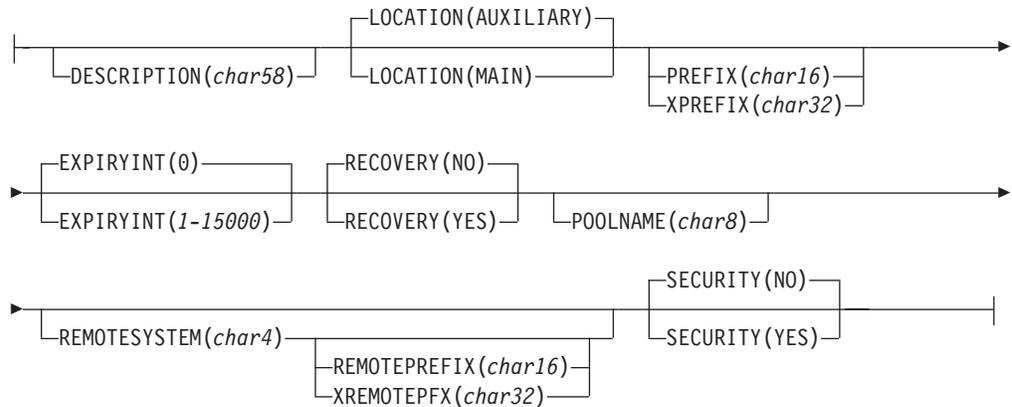
CREATE TSMODEL

Define a model for local, remote, or shared temporary storage queues.

CREATE TSMODEL



CREATE TSMODEL attribute values:



Conditions: INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use `ATTRIBUTES (data-area)` instead of `ATTRIBUTES (data-value)`.

Description

The `CREATE TSMODEL` command installs a `TSMODEL` definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is already a TS model with the name you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

No two TS models can have the same prefix. An attempt to add or replace a model which would result in there being two models with the same prefix will therefore fail.

A syncpoint is implicit in `CREATE TSMODEL` processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the `CREATE` executes successfully, and rolled back if not. See "Creating resource definitions" on page 24 for other general rules governing `CREATE` commands.

Options

ATTRIBUTES (*data-value*)

Specifies the attributes of the TSMODEL resource definition being added. The list of attributes must be coded as a single character string using the syntax shown in **TSMODEL attributes**.

- For details about specific attributes in this resource definition, see TSMODEL attributes.
- For general rules for specifying attributes, see “The ATTRIBUTES option” on page 25.

ATTRLEN (*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length may not exceed 32767 bytes.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

TSMODEL (*data-value*)

Specifies the 8-character name of the TSMODEL definition to be added to the CICS region.

Conditions

INVREQ

RESP2 values:

- n** There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, “RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands,” on page 837 for information on RESP2 values.
- 7** The LOGMESSAGE cvda value is not valid.
- 200** The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1** The length you have specified in ATTRLEN is negative.

NOTAUTH

RESP2 values:

- 100** The user associated with the issuing task is not authorized to use this command.
- 101** The user associated with the issuing task is not authorized to create a TSMODEL definition with this name.

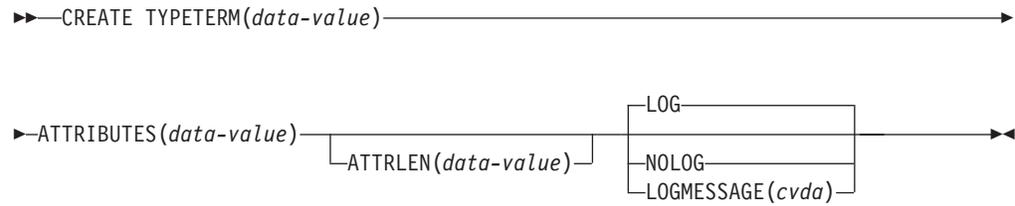
Related information:

 TSMODEL resource definition attributes

CREATE TYPETERM

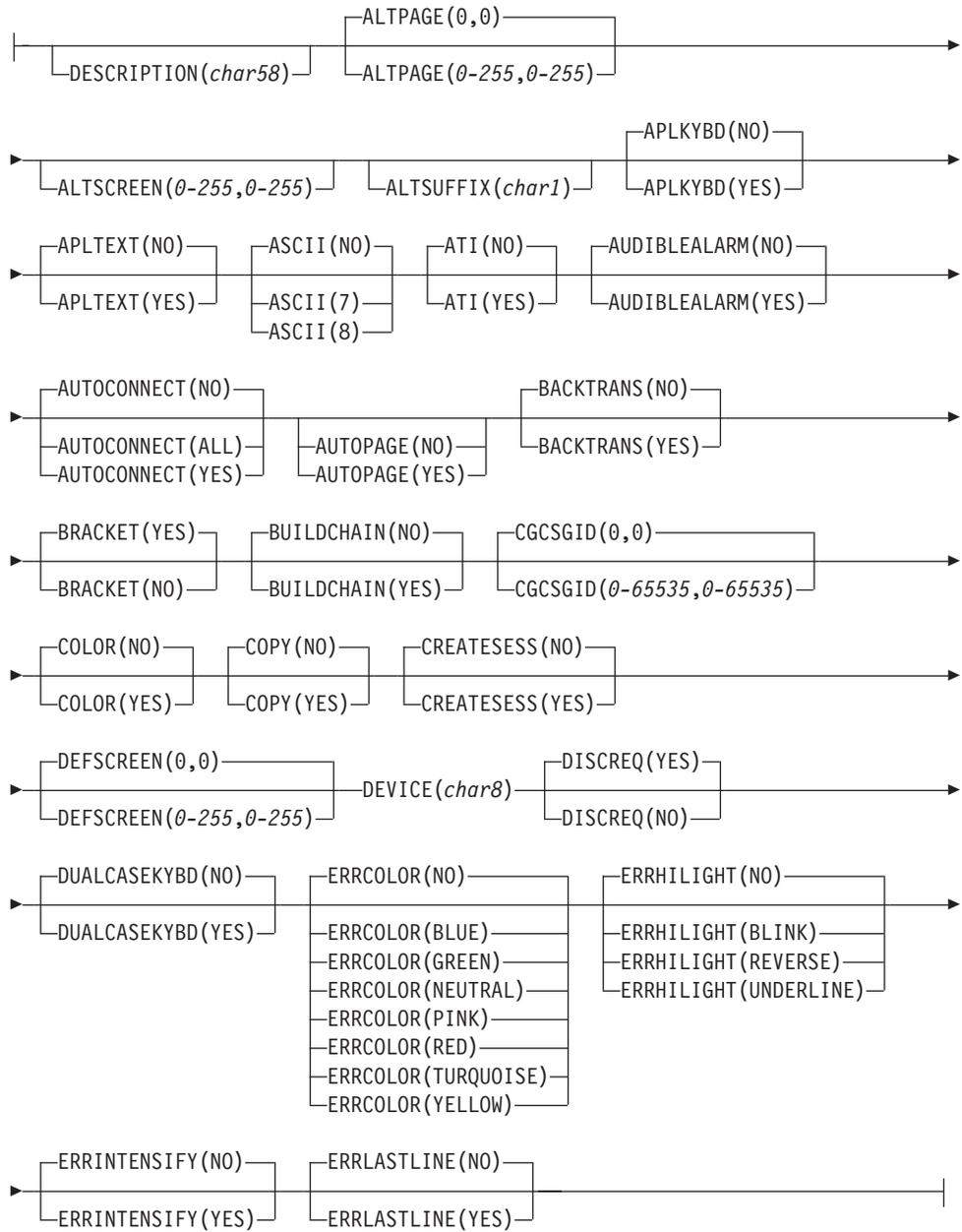
Define a terminal type in the local CICS region.

CREATE TYPETERM

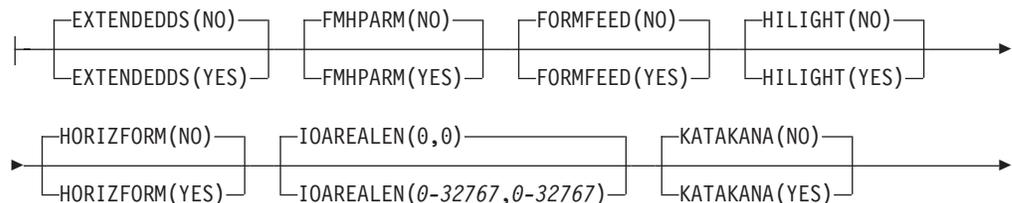


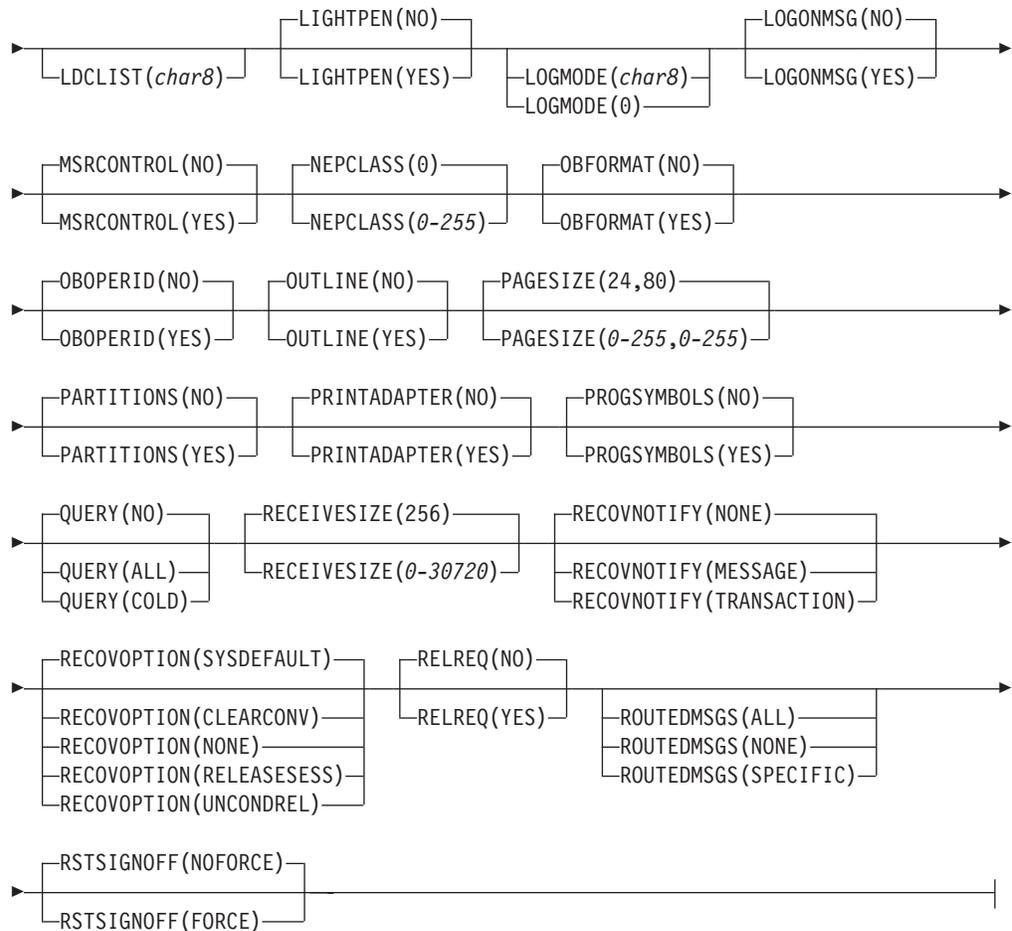
Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

CREATE TYPETERM attribute values (part 1 of 3):

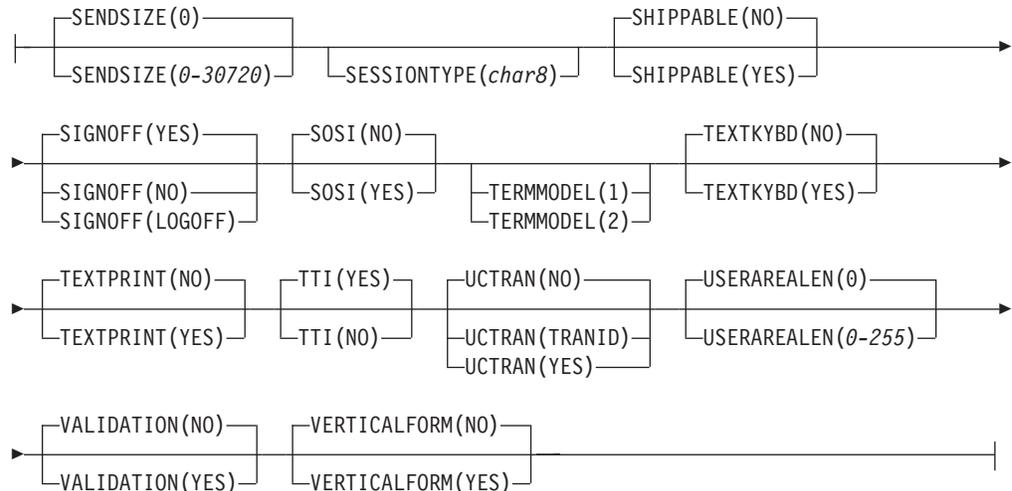


CREATE TYPETERM attribute values (part 2 of 3):





CREATE TYPETERM attribute values (part 3 of 3):



Note to COBOL programmers: In the syntax above, you must use `ATTRIBUTES(data-area)` instead of `ATTRIBUTES(data-value)`.

Description

The CREATE TYPETERM command installs a TYPETERM definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is already a terminal type definition with the name you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

A syncpoint is implicit in CREATE TYPETERM processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not. See “Creating resource definitions” on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES(*data-value*)

specifies the attributes of the TYPETERM being added. The list of attributes must be coded as a single character string using the syntax shown in **TYPETERM attributes**. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN(*data-value*)

specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length may not exceed 32767 bytes.

LOGMESSAGE(*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

TYPETERM(*data-value*)

specifies the 8-character name of the TYPETERM definition to be added to the CICS region.

Conditions

ILLOGIC

RESP2 values:

- 2 The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:

- n There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. See Appendix C, “RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands,” on page 837 for information on RESP2 values.
- 7 The LOGMESSAGE cvda value is not valid.
- 200 The command was executed in a program defined with an

EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1 The length you have specified in ATTRLEN is negative.

NOTAUTH

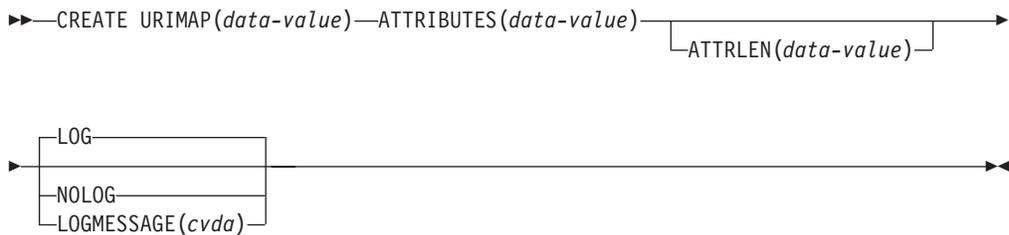
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

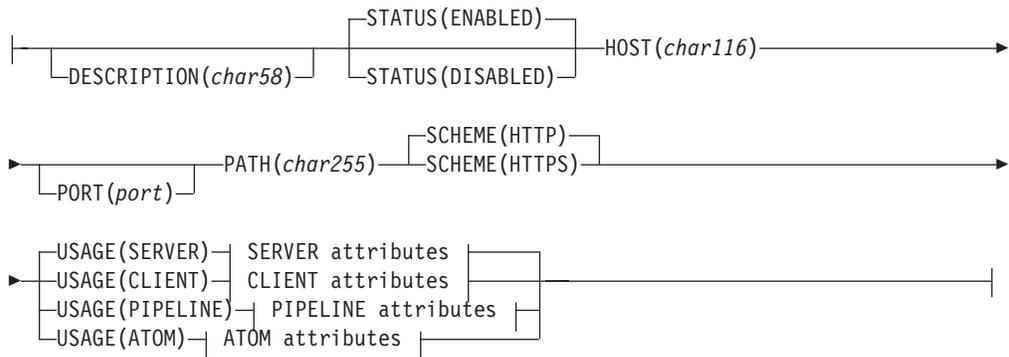
CREATE URIMAP

Define a URIMAP resource in the local CICS region.

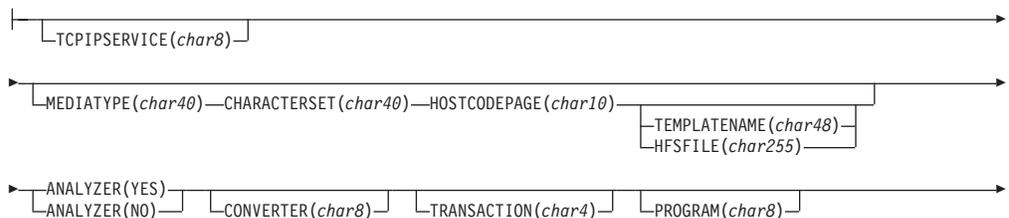
CREATE URIMAP

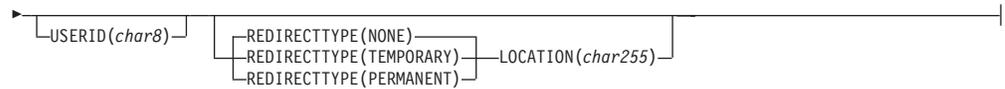


CREATE URIMAP attribute values:

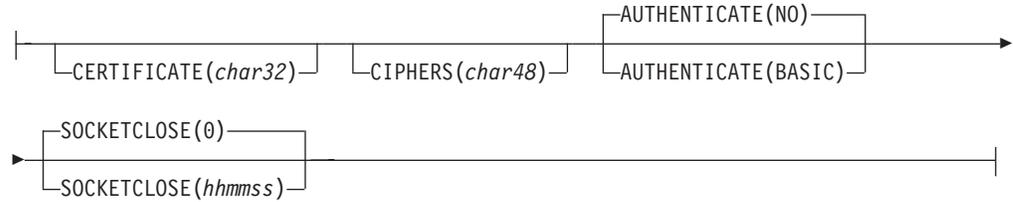


SERVER attributes:

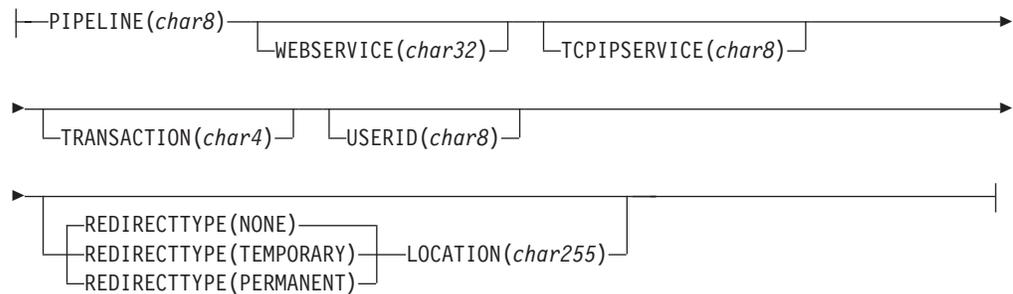




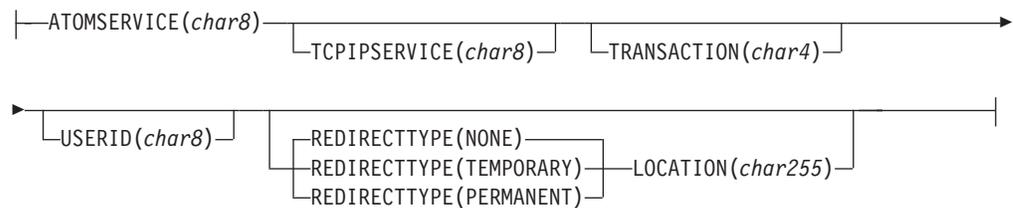
CLIENT attributes:



PIPELINE attributes:



ATOM attributes:



Conditions: INVREQ, LENGERR, NOTAUTH

This command is threadsafe.

Note to COBOL programmers: In the syntax above, you must use **ATTRIBUTES**(data-area) instead of **ATTRIBUTES**(data-value).

Description

The **CREATE URIMAP** command builds a URIMAP resource definition. It does not use a resource definition stored in the CSD. If a URIMAP definition already exists with the name you that specify in the local CICS region, the command fails unless the existing URIMAP definition is disabled, in which case the new definition replaces the old one. If no URIMAP definition with the name specified exists, the new definition is added.

A sync point is implicit in **CREATE URIMAP** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to

recoverable resources made up to that point in the task are committed if the CREATE runs successfully, and rolled back if not. See “Creating resource definitions” on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES (*data-value*)

Specifies the attributes of the URIMAP definition being added. The list of attributes must be coded as a single character string using the syntax shown in URIMAP definition attributes. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN (*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length must not exceed 32,767 bytes.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

URIMAP (*data-value*)

Specifies the 8-character name of the URIMAP definition to be added to the CICS region.

Conditions

INVREQ

RESP2 values:

- n** The ATTRIBUTES string contains a syntax error, or an error occurred during either the discard or resource definition phase of the processing. The RESP2 value is associated with a message written to the transient data queue CSMT, which identifies more precisely the nature of the error. See Appendix C, “RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands,” on page 837 for information on RESP2 values.
- 7** The LOGMESSAGE CVDA value is not valid.
- 200** The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LENGERR

RESP2 values:

- 1** The length that you have specified in ATTRLEN is negative.

NOTAUTH

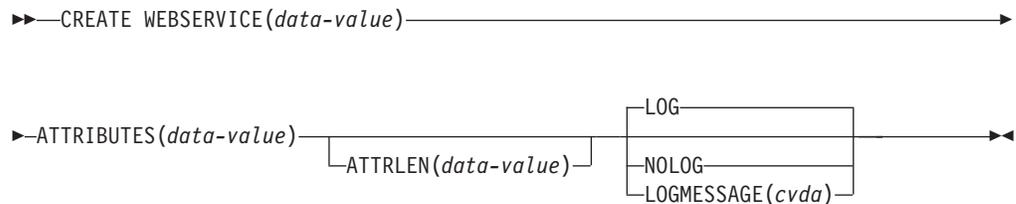
RESP2 values:

- 100** The user associated with the issuing task is not authorized to use this command.

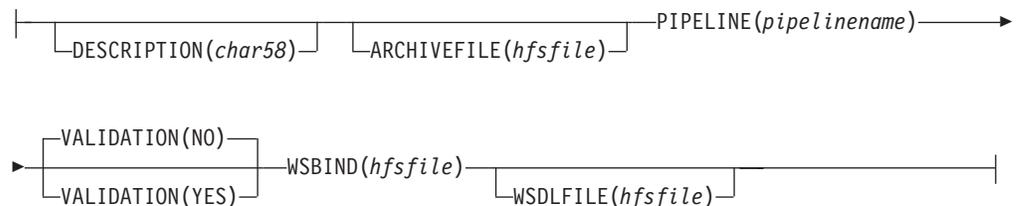
CREATE WEBSERVICE

Define a WEBSERVICE resource in the local CICS region.

CREATE WEBSERVICE



CREATE WEBSERVICE attribute values:



Conditions: INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers: In the syntax above, you must use `ATTRIBUTES(data-area)` instead of `ATTRIBUTES(data-value)`.

Description

The **CREATE WEBSERVICE** command installs a WEBSERVICE resource definition with the attributes specified on the command. It does not use a resource definition stored in the CSD. If a WEBSERVICE resource exists with the name you specify in the local CICS region, the new definition replaces the old one; if not, the new definition is added.

A sync point is implicit in **CREATE WEBSERVICE** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the CREATE executes successfully, and rolled back if not. See “Creating resource definitions” on page 24 for other general rules governing CREATE commands.

Options

ATTRIBUTES(data-value)

Specifies the attributes of the WEBSERVICE resource being added. The list of attributes must be coded as a single character string. See “The ATTRIBUTES option” on page 25 for general rules for specifying attributes, and the *CICS Resource Definition Guide* for details about coding a character string for specific attributes.

ATTRLEN(data-value)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a halfword binary value. The length may not exceed 32767 bytes.

LOGMESSAGE (*cvda*)

Specifies whether CICS logs the attributes used for the resource that is created. CVDA values are as follows:

LOG

The resource attributes are logged to the CSDL transient data queue.

NOLOG

The resource attributes are not logged.

WEBSERVICE (*data-value*)

Specifies the 8-character name of the WEBSERVICE resource definition to be added to the CICS region.

Conditions**INVREQ**

RESP2 values:

- n** There is a syntax error in the ATTRIBUTES string, or an error occurred during either the discard or resource definition phase of the processing. The RESP2 value is associated with a message written to the transient data queue CSMT which identifies more precisely the nature of the error. See Appendix C, "RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands," on page 837 for information on RESP2 values.
- 200** The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.
- 612** Installation of this WEBSERVICE resource failed because it already exists

LENGERR

RESP2 values:

- 1** The length you have specified in ATTRLEN is negative.

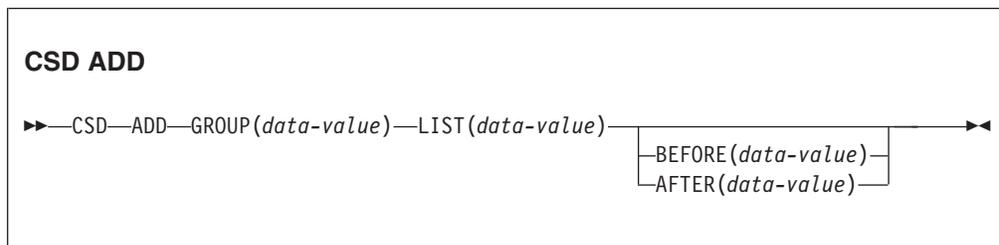
NOTAUTH

RESP2 values:

- 100** The user associated with the issuing task is not authorized to use this command.

CSD ADD

Add a group to a list in the CSD.



Conditions: CSDERR, DUPRES, INVREQ, LOCKED, NOTAUTH, NOTFND

Description

The **CSD ADD** command adds a group to a list, optionally specifying the position in the list using the **BEFORE** or **AFTER** options. If you do not specify **BEFORE** or **AFTER**, the group is added to the end of the list.

A syncpoint is implicit in **CSD ADD** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the command is successful, and rolled back if not.

Options

AFTER(*data-value*)

Specifies the 8-character name of an existing group in the list after which the group is added.

BEFORE(*data-value*)

Specifies the 8-character name of an existing group in the list before which the group is added.

GROUP(*data-value*)

Specifies the 8-character name of the group to be added to the list. You can add a group to a list even if there are no resources in the group.

LIST(*data-value*)

Specifies the 8-character name of the list to which the group is added. If the list does not already exist, a new one is created.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read
- 2 The CSD is READONLY.
- 3 The CSD is full
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

DUPRES

RESP2 value:

- 1 The group already exists in this list.
- 2 The specified group did not exist but a list of the same name is already present in the CSD. The group could not be added
- 3 The specified list did not exist but a group of the same name is already present in the CSD. The list could not be created.

INVREQ

RESP2 values:

- 2 The **GROUP** option contains one or more characters that are not valid.
- 3 The **LIST** option contains one or more characters that are not valid.
- 5 The **BEFORE** option contains one or more characters that are not valid.

- 6 The AFTER option contains one or more characters that are not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LOCKED

RESP 2 values

- 1 The list is locked to another user and cannot be updated.
- 2 The list is IBM-protected.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

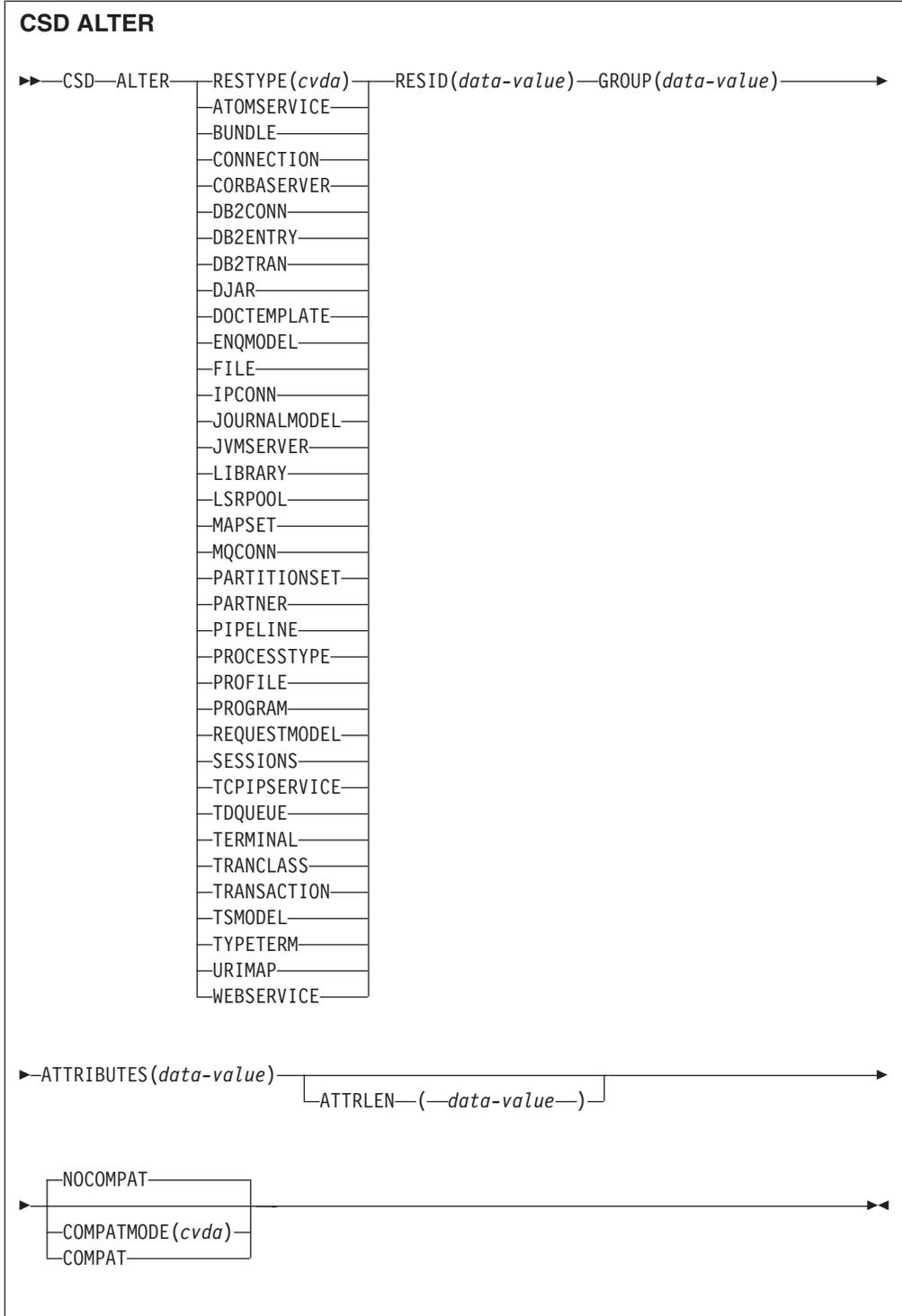
NOTFND

RESP2 value:

- 4 The specified BEFORE or AFTER group does not exist in the list.

CSD ALTER

Change the attributes of an existing resource definition in the CSD.



Conditions: CSDERR, INVREQ, LENGERR, LOCKED, NOTAUTH, NOTFND

Description

The CSD ALTER command changes some or all of the attributes of an existing resource definition in the CSD.

A syncpoint is implicit in **CSD ALTER** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the command is successful, and rolled back if not.

Options

ATTRIBUTES (*data-value*)

Specifies the attributes to be changed. Code the list of attributes as a single character string.

See the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN (*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option, as a fullword binary value.

GROUP (*data-value*)

Specifies the 8-character name of the group containing the resource definition.

RESID (*data-value*)

Specifies the 8-character name of the resource to be altered. Resource names such as TRANSACTION that are only four characters must be padded with four blanks and passed in an 8-character field.

RESTYPE (*cvda*)

Specifies the type of resource to be altered. CVDA values are the resource type names.

COMPATMODE (*cvda*)

Specifies whether obsolete attributes are allowed in the ATTRIBUTES string for this command. Specify one of the following CVDA values:

COMPAT

Obsolete resource attributes are allowed in the ATTRIBUTES string for this command.

NOCOMPAT

Obsolete resource attributes are not allowed in the ATTRIBUTES string for this command.

The default is NOCOMPAT.

Conditions

CSDERR

RESP2 values:

- 1 The CSD cannot be read.
- 2 The CSD is read only.
- 3 The CSD is full.
- 4 The CSD is being used by another CICS system and is not configured for sharing
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

INVREQ

RESP2 values:

- 1 RESTYPE did not specify a valid resource type

- 2 The GROUP option contains one or more characters that are not valid.
- 11 The value of COMPATMODE is not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.
- n The value of ATTRIBUTES or RESID is not valid.

LENGERR

RESP2 value:

- 1 The length specified in ATTRLEN is negative.

LOCKED

RESP2 value:

- 1 The group is locked to another user and cannot be updated.
- 2 The group is IBM-protected.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 value:

- 1 The specified resource definition is not in the named group.
- 2 The named group does not exist.

CSD APPEND

Append the groups in one list on the CSD to the end of another list.

```

CSD APPEND
▶—CSD—APPEND—LIST(data-value)—TO(data-value)—◀
```

Conditions: CSDERR, DUPRES, INVREQ, LOCKED, NOTAUTH, NOTFND

Description

The CSD APPEND command appends the groups in one list on the CSD to the end of another list.

| A syncpoint is implicit in **CSD APPEND** processing, except when an exception
 | condition is detected early in processing the command. Uncommitted changes to
 | recoverable resources made up to that point in the task are committed if the
 | command is successful, and rolled back if not.

Options

LIST(*data-value*)

Specifies the 8-character name of the list that is appended.

TO(*data-value*)

Specifies the 8-character name of the target list that is appended to. The list is created if it does not exist.

Conditions

CSDERR

RESP2 values:

- 1 The CSD could not be read.
- 2 The CSD is read only.
- 3 The CSD is full.
- 4 The CSD is being used by another CICS system and is not configured for sharing
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

DUPRES

RESP2 value:

- 3 Either the list specified in LIST is present in the CSD as a group, or the list specified in TO did not exist but the list cannot be created because a group of the same name is already present in the CSD.

INVREQ

RESP2 values:

- 3 The LIST option contains one or more characters that are not valid.
- 7 The TO option contains one or more characters that are not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LOCKED

RESP2 values:

- 1 The TO list is locked to another user.
- 2 The TO list is IBM protected.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

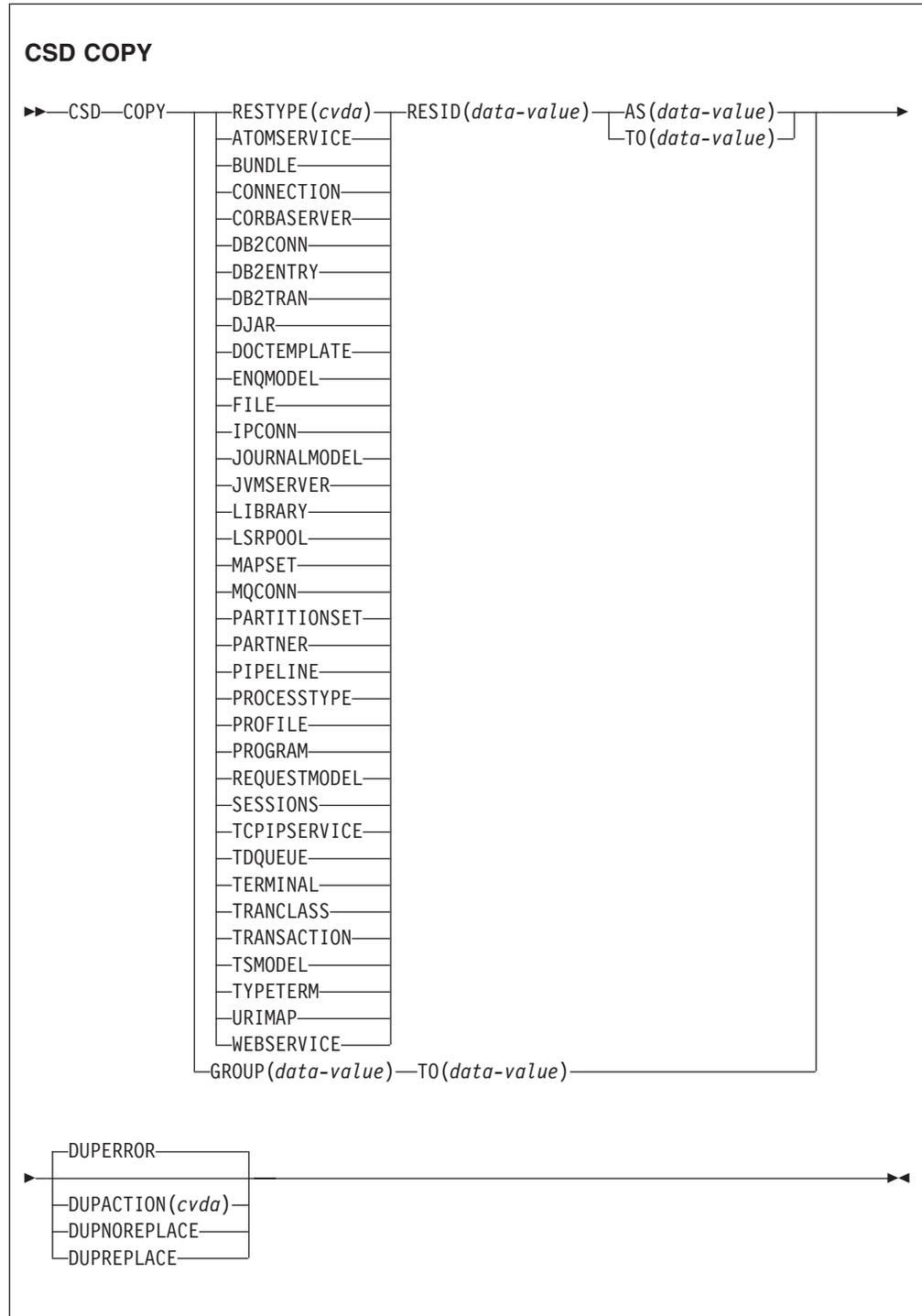
NOTFND

RESP2 value:

- 3 The list specified in the LIST option cannot be found.

CSD COPY

Copy a resource definition in a group to a different group, or copy an entire group.



Conditions: CSDERR, DUPRES, INVREQ, LOCKED, NOTAUTH, NOTFND

Description

The CSD COPY command performs the following operations:

- Copy an individual resource definition to the same group with a new name by using the RESTYPE, RESID and AS options but not TO.
- Copy an individual resource definition to a different group by using the RESTYPE, RESID and TO options.
- Copy an entire group by using the TO option without RESTYPE and RESID.

A syncpoint is implicit in **CSD COPY** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the command is successful, and rolled back if not.

Options

AS(*data-value*)

Specifies the new 8-character name of an individual resource definition. For resources with 4-character names, the first four characters of this value are used.

GROUP(*data-value*)

Specifies the 8-character name of the group to be copied, or the group that contains the individual resource definition to be copied.

DUPACTION(*cvda*)

Specifies the required action when there are duplicate definitions in the target group. The default value is DUPERROR. CVDA values are as follows:

DUPERROR

Raises the DUPRES condition for duplicate definitions.

DUPNOREPLACE

Specifies that duplicate definitions in the target group are not replaced.

DUPREPLACE

Specifies that duplicate definitions in the target group are replaced.

RESID(*data-value*)

Specifies the 8-character name of the individual resource definition to be copied. Resource names such as TRANSACTION that are only four characters must be padded with four blanks and passed in an 8-character field.

RESTYPE(*cvda*)

Specifies the type of resource definition to be copied. CVDA values are the resource type names.

TO Specifies the 8-character name of the group to which the individual resource definition or whole group is to be copied. If an individual resource definition is specified and the TO option is not, the resource definition is copied in the same group. In this case you must specify the AS option. You must use the TO option if a whole group is to be copied. In all cases, the TO group is created if it does not exist.

Conditions

CSDERR

RESP2 value:

- 1 The CSD could not be read.
- 2 The CSD is read only.

- 3 The CSD is full.
- 4 The CSD is being used by another CICS system and is not configured for sharing
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

DUPRES

RESP2 value:

- 1 The group exists already (for a whole group copy) or one or more of the resource definitions to be created by the COPY already exists and DUPACTION was set or defaulted to DUPERROR.
- 2 The name specified in the GROUP or TO option is present in the CSD as a list.

INVREQ

RESP2 values:

- 1 The resource type specified for RESTYPE is not valid.
- 2 The GROUP option contains one or more characters that are not valid.
- 4 The RESID option contains one or more characters that are not valid.
- 7 The TO option contains one or more characters that are not valid.
- 9 The DUPACTION value is not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LOCKED

RESP2 values:

- 1 The target group is locked to another user.
- 2 The target group is IBM protected.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

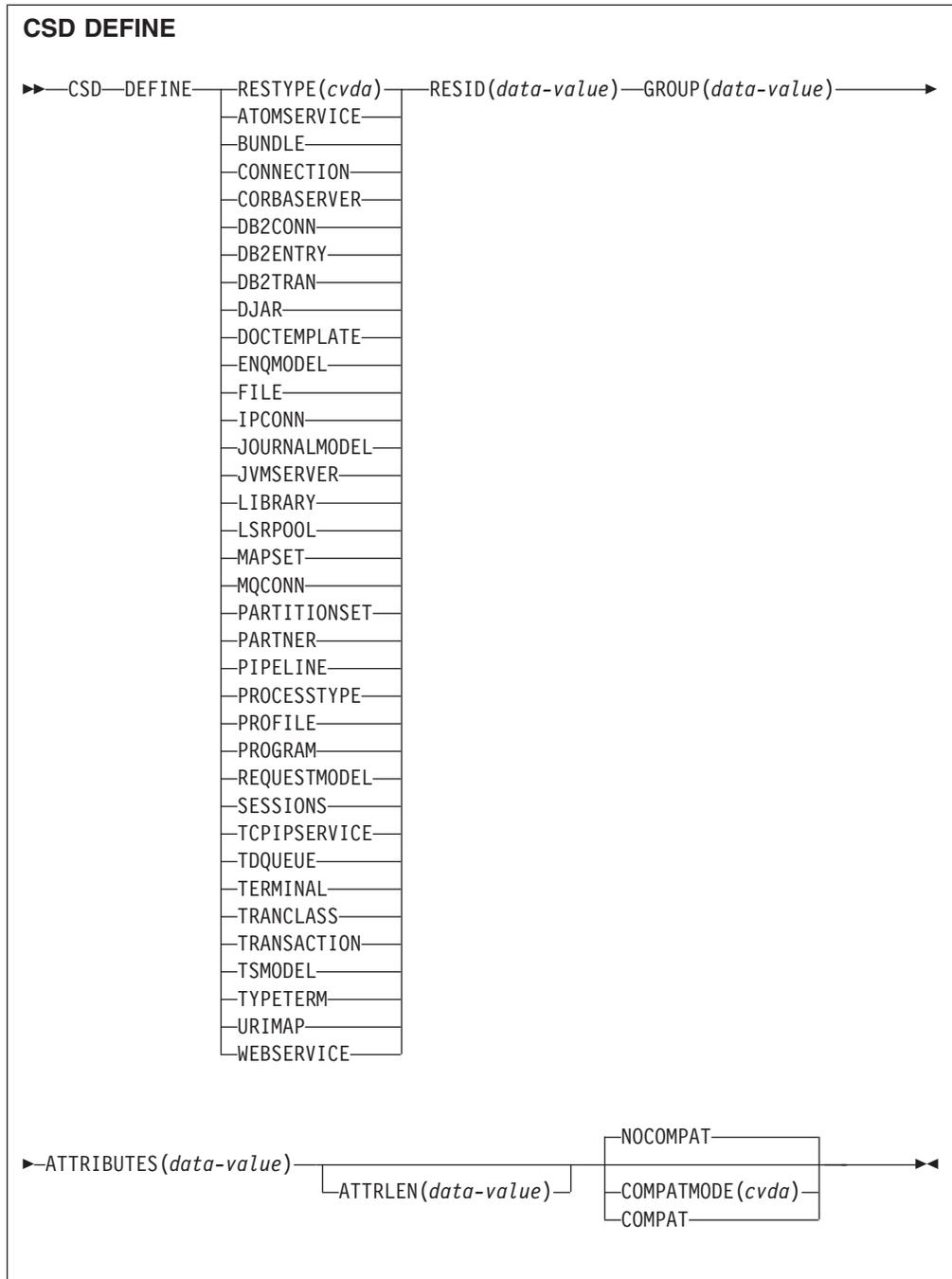
NOTFND

RESP2 values:

- 1 The specified resource definition cannot be found.
- 2 The source group cannot be found.

CSD DEFINE

Create a new resource definition in the CSD.



Conditions: CSDERR, DUPRES, INVREQ, LENGERR, LOCKED, NOTAUTH

Description

The CSD DEFINE command creates a new resource definition on the CSD.

| A syncpoint is implicit in **CSD DEFINE** processing, except when an exception
 | condition is detected early in processing the command. Uncommitted changes to
 | recoverable resources made up to that point in the task are committed if the
 | command is successful, and rolled back if not.

Options

ATTRIBUTES (*data-value*)

Specifies the attributes of the new resource. Code the list of attributes as a single character string.

See the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN (*data-value*)

Specifies the length in bytes of the character string supplied in the ATTRIBUTES option as a fullword binary value.

GROUP (*data-value*)

Specifies the 8-character name of the group containing the resource definition.

RESID (*data-value*)

Specifies the 8-character name of the resource to be defined. Resource names such as TRANSACTION that are only four characters in length must be padded with four blanks and passed in an 8-character field.

RESTYPE (*cvda*)

Specifies the type of resource definition to be defined. CVDA values are the resource type names.

COMPATMODE (*cvda*)

Specifies whether obsolete attributes are allowed in the ATTRIBUTES string for this command. Specify one of the following CVDA values:

COMPAT

Obsolete resource attributes are allowed in the ATTRIBUTES string for this command.

NOCOMPAT

Obsolete resource attributes are not allowed in the ATTRIBUTES string for this command.

The default is NOCOMPAT.

Conditions

CSDERR

RESP2 values:

- 1 The CSD cannot be read.
- 2 The CSD is read only.
- 3 The CSD is full.
- 4 The CSD is being used by another CICS system and is not configured for sharing
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

DUPRES

RESP2 values:

- 1 A resource of this name and type already exists in the specified group.
- 2 The specified group did not exist but because a list of the same name is already present in the CSD, the group could not be created.

INVREQ

RESP2 values:

- 1 The resource type specified for RESTYPE is not valid.
- 2 The GROUP option contains one or more characters that are not valid.
- 11 The value of COMPATMODE is not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.
- n The ATTRIBUTES string contains a syntax error or RESID contains a character that is not valid.

LENGERR

RESP2 value:

- 1 The length specified in ATTRLEN is negative.

LOCKED

RESP2 values:

- 1 The specified group is locked to another user.
- 2 The group is IBM-protected.

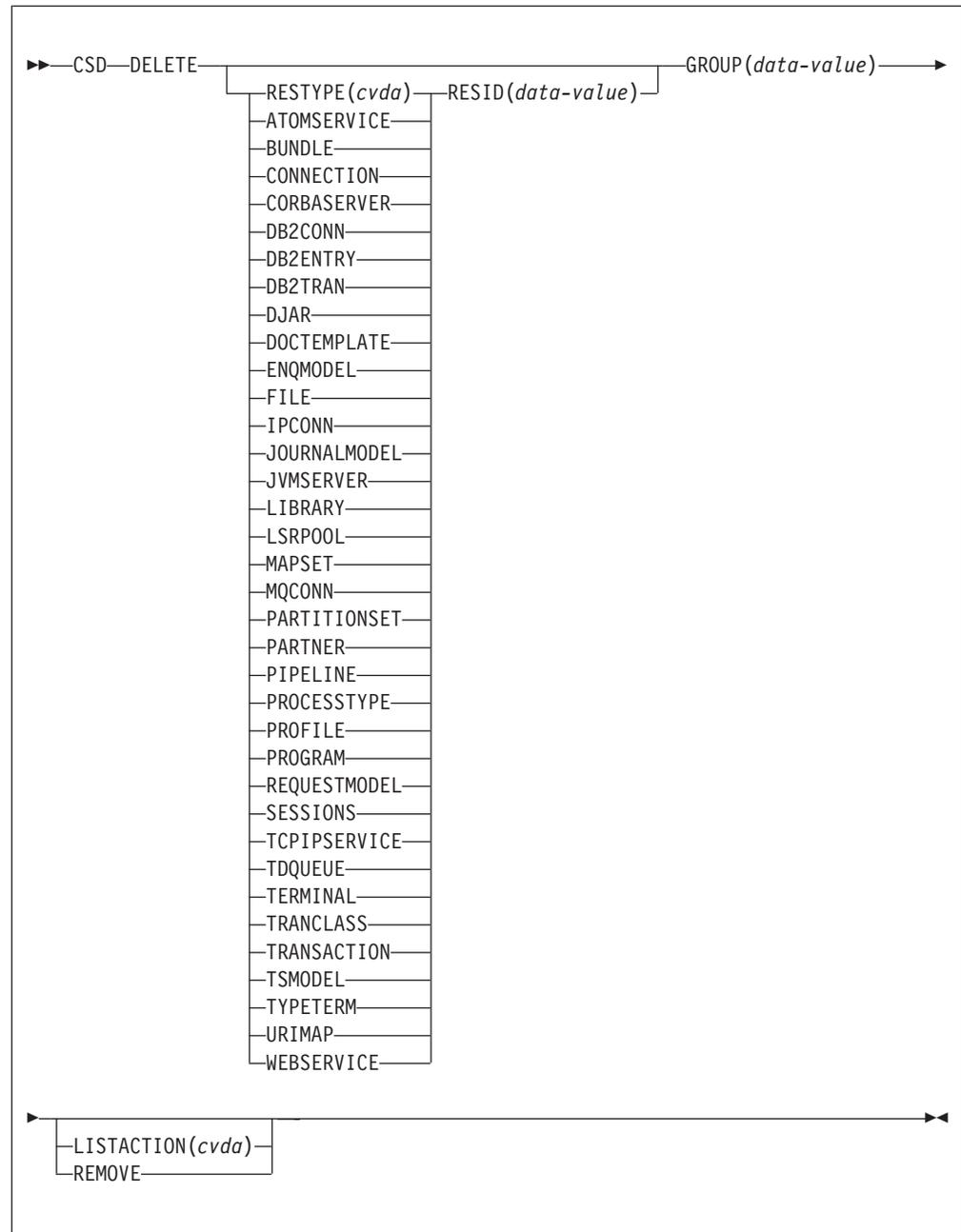
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD DELETE

Delete a group, or a single resource definition in a group, from the CSD.



Conditions: CSDERR, INVREQ, LOCKED, NOTAUTH, NOTFND

Description

The CSD DELETE command performs the following operations:

- Delete a single resource from a group in the CSD.
- Delete a whole group from the CSD.

A syncpoint is implicit in **CSD DELETE** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the command is successful, and rolled back if not.

Options

GROUP(*data-value*)

Specifies the 8-character name of the group to be deleted, or the group containing the resource definition to be deleted.

LISTACTION(*cvda*)

Specifies the effect that group delete has on lists that contain the group. CVDA value is:

REMOVE

The group is removed from all lists that contain it.

RESID(*data-value*)

Specifies the 8-character name of the resource definition to be deleted. Resource names such as TRANSACTION that are only four characters in length must be padded with four blanks and passed in an 8-character field.

RESTYPE(*cvda*)

Specifies the type of resource to be deleted. CVDA values are the resource type names.

Conditions

CSDERR

RESP2 values:

- 1 The CSD cannot be read.
- 2 The CSD is read only.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

INVREQ

RESP2 values:

- 1 The resource type specified for RESTYPE is not valid.
- 2 The GROUP option contains one or more characters that are not valid.
- 4 The RESID option contains one or more characters that are not valid.
- 10 The value of LISTACTION is not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LOCKED

RESP2 value:

- 1 The specified group is locked to another user.
- 2 The group is IBM-protected.

NOTAUTH

RESP2 value:



Conditions: CSDERR, NOTAUTH

Description

The ENDBRGROUP command stops a browse of groups in the CSD started by a CSD STARTBRGROUP command. The browse can be of all the groups in the CSD, or of all the groups in a specified list.

Options

LIST

Specifies that the browse being ended is of the groups in a list rather than all groups in the CSD.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be accessed.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

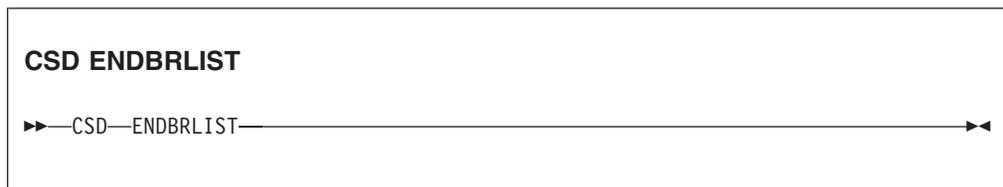
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD ENDBRLIST

End the current browse of the lists in the CSD.



Conditions: CSDERR, NOTAUTH

Description

The CSD ENDBRLIST command stops a browse of the lists in the CSD started by a CSD STARTBRLIST command.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be accessed.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

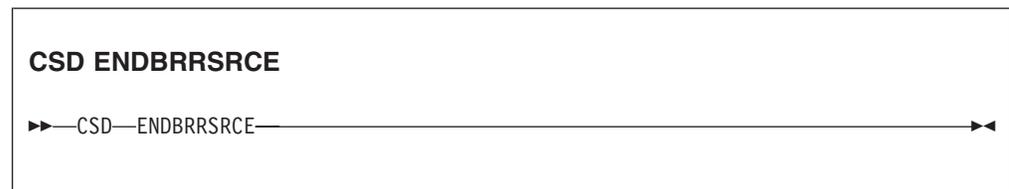
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD ENDBRRSRCE

End the current browse of the resources in a specified group.



Conditions: CSDERR, NOTAUTH

Description

The **CSD ENDBRRSRCE** command stops a browse of the resource definitions in a group in the CSD that was started by a **CSD STARTBRRSRCE** command.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be accessed.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

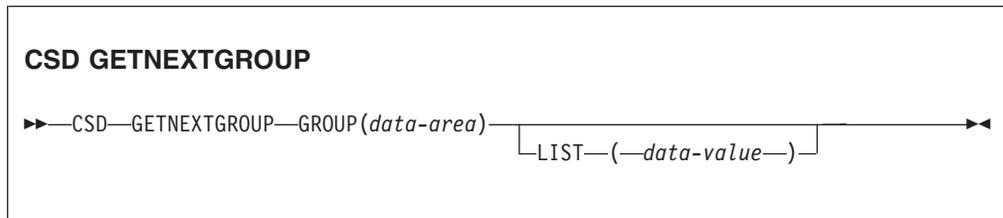
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD GETNEXTGROUP

Get the next group in a group browse.



Conditions: CSDERR, END, ILLOGIC, NOTAUTH

Description

The **CSD GETNEXTGROUP** command returns the name of the next group in the browse started by a **CSD STARTBRGROUP** command.

Options

GROUP (*data-area*)

Returns the 8-character name of the group.

LIST (*data-value*)

Specifies the 8-character name of the list to which the browse was limited on the STARTBRGROUP command. You must specify a value for LIST if the associated STARTBRGROUP command includes the LIST option.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

END

RESP2 value:

- 1 The CSD or the list contain no more groups.

ILLOGIC

RESP2 value:

- 1 A group browse is not in progress.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD GETNEXTLIST

Get the next list in a list browse.

CSD GETNEXTLIST

►—CSD—GETNEXTLIST—LIST(*data-area*)—◄

Conditions: CSDERR, END, ILLOGIC, NOTAUTH

Description

The **CSD GETNEXTLIST** command returns the name of the next list in a browse started by a **CSD STARTBRLIST** command.

Options

LIST(*data-area*)

Returns the 8-character name of the list.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

END

RESP2 value:

- 1 The CSD contains no more lists.

ILLOGIC

RESP2 value:

- 1 A list browse is not in progress.

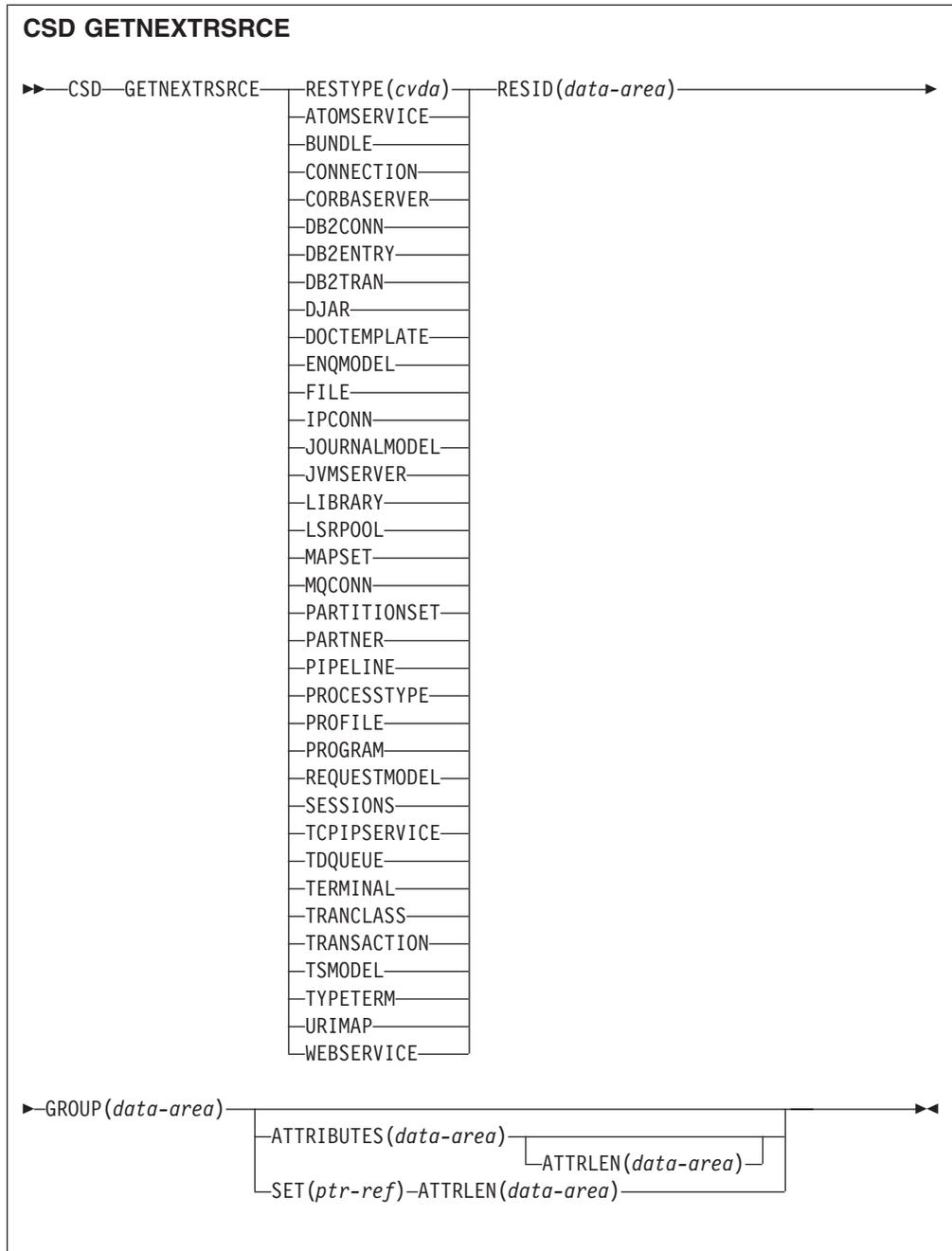
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD GETNEXTSRCE

Get the details of the next resource in a resource browse.



Conditions: CSDERR, END, ILLOGIC, LENGERR, NOTAUTH

Description

The **CSD GETNEXTRSRCE** command returns the details of the next resource in a browse started by a **CSD STARTBRRSRCE** command.

Options

ATTRIBUTES(*data-area*)

Specifies the data area in which a character string containing a list of attributes of the relevant resource is returned.

See the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN(*data-area*)

A fullword binary field containing one of the following values:

- When used with the ATTRIBUTES option:
 - On input ATTRLEN contains the maximum length of the attributes string that the application can accept. You do not need to specify ATTRLEN if the length can be generated by the compiler from the ATTRIBUTES variable.
 - On output, ATTRLEN contains the length of the attributes string returned. The LENGERR condition is raised if the attribute string is longer than the input ATTRLEN value.
- When used with the SET option, ATTRLEN is an output-only field that is set to the length of the attributes string.

GROUP(*data-area*)

Returns the 8-character name of the group being browsed.

RESID(*data-area*)

Returns the 8-character name of the resource definition whose attributes are returned. Resource names such as TRANSACTION that are only four characters are padded with four blanks.

RESTYPE(*cvda*)

Returns the type of the resource definition. CVDA values are the resource type names.

SET(*ptr-ref*)

Specifies a pointer reference that is set to the address of the returned attributes string. The pointer reference is valid until the next CSD GETNEXTRSRCE resource command is issued, or until the end of the task.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

END

RESP2 value:

- 1 The CSD or the specified list contains no more groups.

ILLOGIC

RESP2 value:

- 1 No resource browse is in progress.

LENGERR

RESP2 value:

- 1 The length of the ATTRIBUTES data area as specified on the ATTRLEN option passed to CICS is negative.
- 2 The length of the ATTRIBUTES data area as specified on the ATTRLEN option passed to CICS is less than the amount of data to be returned.

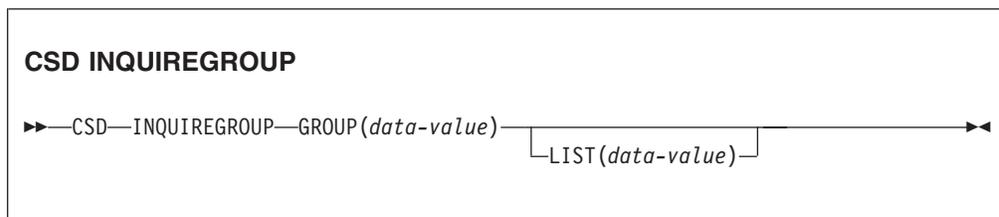
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD INQUIREGROUP

Inquire on a group in the CSD or on a group in a specified list in the CSD.



Conditions:CSDERR, NOTAUTH, NOTFND

Description

Use the **CSD INQUIREGROUP** command to make a direct inquiry of group names on the CSD. You can limit the scope of the inquiry to a specified list. If the response is *NORMAL*, the specified group exists on the CSD or in the list.

Options

GROUP(data-value)

Specifies the 8-character name of the group being queried.

LIST(data-value)

Specifies the 8-character name of a list to which the scope of the query is limited.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

- 2 The specified group cannot be found.
- 3 The specified list cannot be found.

CSD INQUIRELIST

Inquire on a list in the CSD.

CSD INQUIRELIST

▶—CSD—INQUIRELIST—LIST(*data-value*)—▶

Conditions: CSDERR, NOTAUTH, NOTFND

Description

The **CSD INQUIRELIST** command makes a direct inquiry of list names on the CSD. If the response is **NORMAL**, the specified group exists on the CSD.

Options

LIST(*data-value*)

Specifies the 8-character name of the list being queried.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

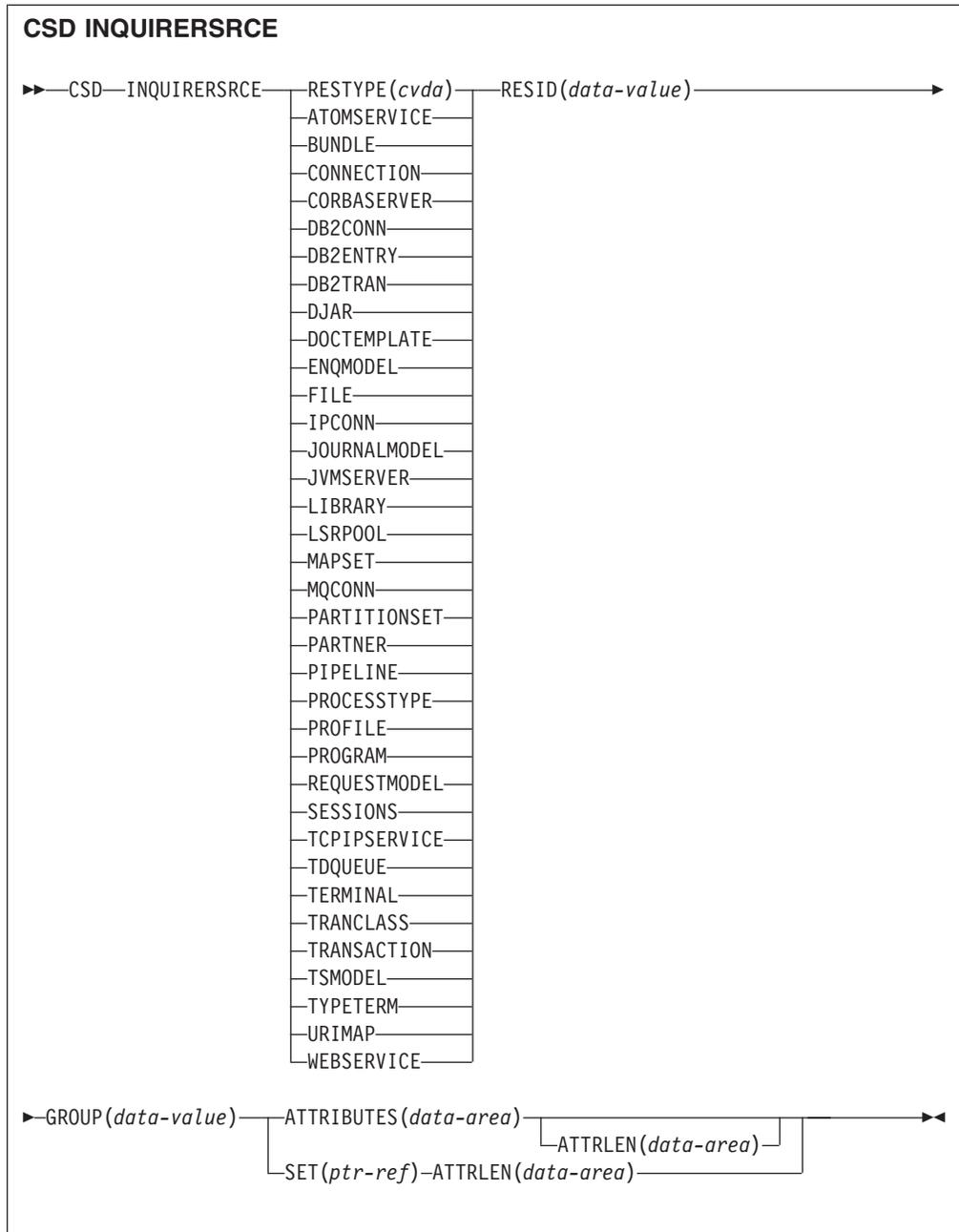
NOTFND

RESP2 value:

- 3 The specified list cannot be found.

CSD INQUIRERSRCE

Inquire on the attributes of a resource in a specified group in the CSD.



Conditions: CSDERR, INVREQ, LENGERR, NOTAUTH, NOTFND

Description

The **CSD INQUIRERSRCE** command queries the attributes of resources in a specified group on the CSD.

Options

ATTRIBUTES (data-area)

Specifies the data area in which a character string containing a list of attributes of the relevant resource is returned.

See the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN(*data-area*)

A fullword binary field containing one of the following values:

- When used with the ATTRIBUTES option:
 - On input, ATTRLEN contains the maximum length of the attributes string that the application can accept. You do not have to specify ATTRLEN if the length can be generated by the compiler from the ATTRIBUTES variable.
 - On output, ATTRLEN contains the length of the attributes string returned. The LENGERR condition is raised if the attribute string is longer than the input ATTRLEN value.
- When used with the SET option, ATTRLEN is an output-only field that is set to the length of the attributes string.

GROUP(*data-value*)

Returns the 8-character name of the group containing the resource definition or resource definition attributes being queried.

RESID(*data-value*)

Returns the 8-character name of the resource definition whose attributes are being queried. Resource names such as TRANSACTION that are only four characters in length must be padded with four blanks.

RESTYPE(*cvda*)

Returns the resource type of the resource definition being queried. CVDA values are the resource type names.

SET(*ptr-ref*)

Specifies a pointer reference that is set to the address of the returned attributes string. The pointer reference is valid until the next CSD INQUIRERSRCE resource command is issued or until the end of the task.

Conditions**CSDERR**

RESP2 value:

- 1 The CSD cannot be read.
- 2 The CSD is defined as readonly.
- 3 There is no more space available in the CSD.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

INVREQ

RESP2 values:

- 1 The specified RESTYPE is not a valid resource type.
- 4 The RESID option contains one or more characters that are not valid for the specific resource type.

LENGERR

RESP2 value:

- 1 The length of the ATTRIBUTES data area as specified on the ATTRLEN option passed to CICS is negative.

- 2 The length of the ATTRIBUTES data area as specified on the ATTRLEN option passed to CICS is less than the amount of data to be returned.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 value:

- 1 The specified resource definition cannot be found.
- 2 The specified group cannot be found.

CSD INSTALL

Install a list, a group, or a single resource definition in a group, from the CSD.

CSD INSTALL syntax for a list

▶▶—CSD—INSTALL—LIST(*data-value*)—————▶▶

A syncpoint is implicit in **CSD INSTALL** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the command is successful, and rolled back if not.

Options

GROUP(*data-value*)

Specifies the 8-character name of the group to be installed, or the group containing the individual resource definition to be installed.

LIST(*data-value*)

Specifies the 8-character name of the list to be installed.

RESID(*data-value*)

Specifies the 8-character name of the individual resource definition to be installed. Resource names such as TRANSACTION that are only 4 characters in length must be padded with 4 blanks and passed in an 8-character field.

RESTYPE(*cvda*)

Specifies the type of resource to be installed. CVDA values are the resource type names.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

INCOMPLETE

RESP2 value:

- 1 The install of a complete group or list was only partially successful.

INVREQ

RESP2 values:

- 1 The resource type specified for RESTYPE is not valid.
- 2 The GROUP option contains one or more characters that are not valid.
- 3 The LIST option contains one or more characters that are not valid.
- 4 The RESID option contains one or more characters that are not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET, or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.
- n An error occurred while adding the specified resource to the running system due to the current state of the system.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

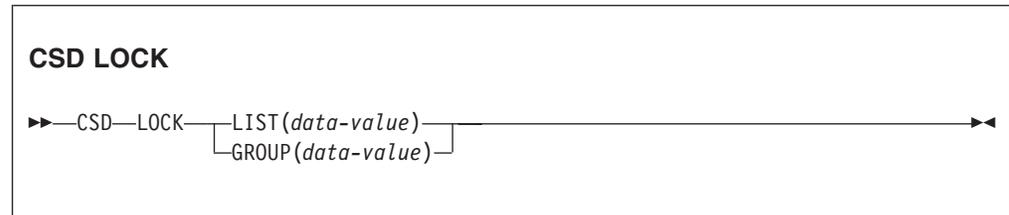
NOTFND

RESP2 values:

- 1 The specified resource definition cannot be found.
- 2 The specified group cannot be found.
- 3 The specified list cannot be found.

CSD LOCK

Restrict update and delete access for a group or list to a single operator identifier.



Conditions: CSDERR, DUPRES, INVREQ, LOCKED, NOTAUTH

Description

When you lock a group or list, other users can view or copy it but they are restricted from changing or deleting it. You can lock a nonexistent group or list to reserve the named group or list for your own future use. The only command that releases a lock is the **UNLOCK** command. No other RDO commands can unlock a group or list. For example, if you delete all the resources in a group, or all the groups in a list, the lock remains.

You use the **LOCK** and **UNLOCK** commands to control update access to a group or list so that only operators with the same operator identifier can make changes.

Users who are not signed on or who have a different operator identifier (OPIDENT) are not allowed to perform any operation that changes the locked list or group. However, any user is allowed to perform the following operations on a locked group or list:

- CHECK (CEDA)
- COPY
- DISPLAY (CEDA)
- INQUIRE/BROWSE
- INSTALL
- VIEW (CEDA)

Only a user on the same system and with the same operator identifier can remove the lock, using the UNLOCK command.

A syncpoint is implicit in **CSD LOCK** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the command is successful, and rolled back if not.

Options

GROUP (*data-value*)

Specifies the 8-character name of the group to be locked.

LIST (*data-value*)

Specifies the 8-character name of the list to be locked.

Conditions

CSDERR

RESP2 values:

- 1 The CSD cannot be read.
- 2 The CSD is read only.
- 3 The CSD is full.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

DUPRES

RESP2 values:

- 2 The name specified in **GROUP** exists in the CSD as a list.
- 3 The name specified in **LIST** exists in the CSD as a group.

INVREQ

RESP2 values:

- 2 The **GROUP** option contains one or more characters that are not valid.
- 3 The **LIST** option contains one or more characters that are not valid.
- 200 The command was run in a program defined with an **EXECUTIONSET** value of **DPLSUBSET** or in a program invoked from a remote system by a distributed program link without the **SYNCONRETURN** option.

LOCKED

RESP2 values:

- 1 The group or list is already locked to another user.
- 2 The group or list is IBM-protected.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD REMOVE

Remove a group from a list in the CSD.

CSD REMOVE

▶—CSD—REMOVE—GROUP(*data-value*)—LIST(*data-value*)—▶

Conditions: CSDERR, INVREQ, LOCKED, NOTAUTH, NOTFND

Description

The **CSD REMOVE** command removes a group from a list.

The group and all its resource definitions still exist in the CSD file. When the last group is removed from a list, the list no longer exists in the CSD file.

When a group is deleted, you can request that the group is removed from all lists that contained it. When the last group is removed from a list, the list is deleted.

A syncpoint is implicit in **CSD REMOVE** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the command is successful, and rolled back if not.

Options

GROUP(*data-value*)

Specifies the 8-character name of the group to be removed.

LIST(*data-value*)

Specifies the 8-character name of the list from which the group is to be removed.

Conditions

CSDERR

RESP2 values:

- 1 The CSD cannot be read.
- 2 The CSD is read only.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

INVREQ

RESP2 values:

- 2 The GROUP option contains one or more characters that are not valid.
- 3 The LIST option contains one or more characters that are not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LOCKED

RESP2 values:

- 1 The list is locked to another user and cannot be updated.

2 The list is IBM-protected.

NOTAUTH

RESP2 value:

100 The user associated with the issuing task is not authorized to use this command.

NOTFND

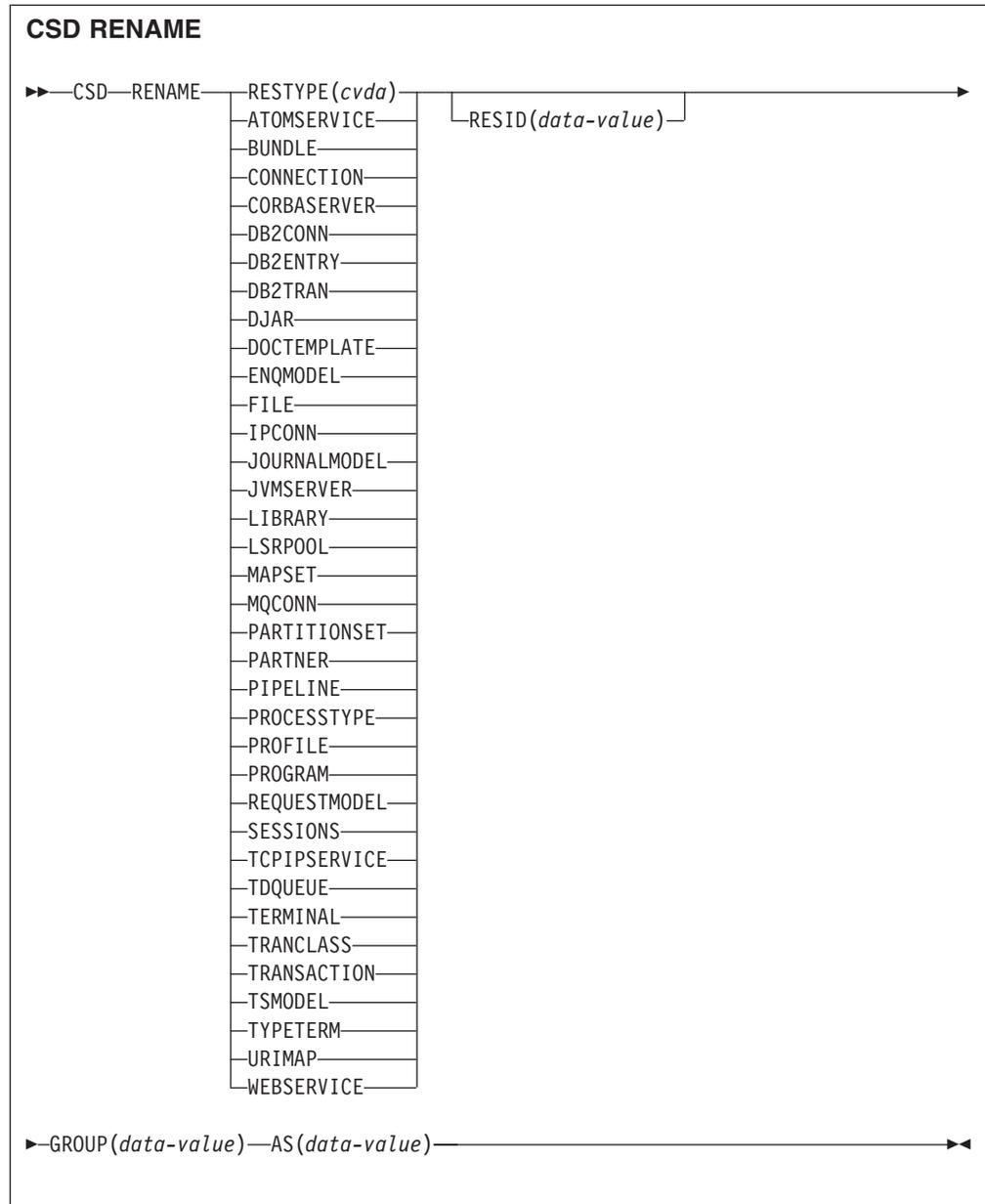
RESP2 values:

2 The specified group cannot be found.

3 The specified list cannot be found.

CSD RENAME

Rename a resource definition in the CSD.



Conditions: CSDERR, DUPRES, INVREQ, LOCKED, NOTAUTH, NOTFND

Description

The **CSD RENAME** command renames an individual resource definition in a specified group.

|
|
|
|

A syncpoint is implicit in **CSD RENAME** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the command is successful, and rolled back if not.

Options

AS(*data-value*)

Specifies the new 8-character name of the resource definition. For resources with 4-character names, the first four characters of this value are used.

GROUP(*data-value*)

Specifies the 8-character name of the group containing the resource definition to be renamed.

RESID(*data-value*)

Specifies the 8-character name of the resource to be renamed. Resource names that are only four characters in length must be padded with four blanks and passed in an 8-character field.

Conditions

CSDERR

RESP2 values:

- 1 The CSD cannot be read.
- 2 The CSD is read only.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

DUPRES

RESP2 value:

- 1 The specified resource definition already exists.

INVREQ

RESP2 values:

- 1 The resource type specified for RESTYPE is not valid.
- 2 The GROUP option contains one or more characters that are not valid.
- 4 The RESID or AS option contains one or more characters that are not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSET or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

LOCKED

RESP2 values:

- 1 The specified group is already locked to another user.
- 2 The group is IBM-protected.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

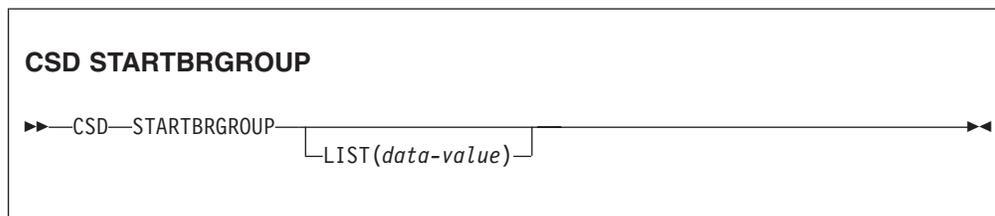
NOTFND

RESP2 values:

- 1 The specified resource definition cannot be found.

CSD STARTBRGROUP

Start a browse of the groups in the CSD or of the groups in a list.



Conditions: CSDERR, ILLOGIC, NOTAUTH, NOTFND

Description

The **CSD STARTBRGROUP** command starts a CSD group browse. The browse can be of all the groups in the CSD, or of all the groups in a specified list.

Rules for concurrent browses

There are four distinct types of browse that can be used on the CSD:

1. LIST browse of all the lists in the CSD.
2. ALL GROUPS browse of all the groups in the CSD.
3. GROUPS IN LIST browse of all the groups in a specified list.
4. RESOURCE browse of all resources in a specified group.

The rules that govern concurrent browses in the same transaction are as follows:

- Browses of the same type are not allowed together.
- The LIST and ALL GROUPS browse are not allowed together.
- The GROUPS IN LIST and RESOURCE browses are not allowed together.

The ILLOGIC condition is raised if these rules are broken.

For example, it is permitted to browse all groups in the CSD and, as each one is returned, browse all of the resources in that group.

Options

LIST(*data-value*)

Specifies the 8-character name of a list to which the browse is to be limited. If you do not specify this option, all groups in the CSD are returned on the browse.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.

- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

ILLOGIC

RESP2 value:

- 2 A browse of the same type or a conflicting type is already in progress.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 value:

- 3 The specified list cannot be found.

CSD STARTBRLIST

Start a browse of the lists in the CSD.



Conditions: CSDERR, ILLOGIC, NOTAUTH,

Description

The **CSD STARTBRLIST** command starts a browse of all the lists in the CSD.

Rules for concurrent browses

There are four distinct types of browse that can be used on the CSD:

1. LIST browse of all the lists in the CSD.
2. ALL GROUPS browse of all the groups in the CSD.
3. GROUPS IN LIST browse of all the groups in a specified list.
4. RESOURCE browse of all resources in a specified group.

The rules that govern concurrent browses in the same transaction are as follows:

- Browses of the same type are not allowed together.
- The LIST and ALL GROUPS browse are not allowed together.
- The GROUPS IN LIST and RESOURCE browses are not allowed together.

The ILLOGIC condition is raised if these rules are broken.

For example, it is permitted to browse all groups in the CSD and, as each one is returned, browse all of the resources in that group.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

ILLOGIC

RESP2 value:

- 2 A list browse or a conflicting browse is already in progress.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD STARTBRRSRCE

Start a browse of the resources in a specified group.

CSD STARTBRRSRCE

►►—CSD—STARTBRRSRCE—GROUP(*data-value*)—————►►

Conditions: CSDERR, ILLOGIC, NOTAUTH, NOTFND

Description

The **CSD STARTBRRSRCE** command starts a browse of all the resource definitions in a specified group.

Rules for concurrent browses

There are four distinct types of browse that can be used on the CSD:

1. LIST browse of all the lists in the CSD.
2. ALL GROUPS browse of all the groups in the CSD.
3. GROUPS IN LIST browse of all the groups in a specified list.
4. RESOURCE browse of all resources in a specified group.

The rules that govern concurrent browses in the same transaction are as follows:

- Browses of the same type are not allowed together.
- The LIST and ALL GROUPS browse are not allowed together.
- The GROUPS IN LIST and RESOURCE browses are not allowed together.

The ILLOGIC condition is raised if these rules are broken.

For example, it is permitted to browse all groups in the CSD and, as each one is returned, browse all of the resources in that group.

Options

GROUP (*data-value*)

Specifies the 8-character name of the group to be browsed.

Conditions

CSDERR

RESP2 value:

- 1 The CSD cannot be read.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

ILLOGIC

RESP2 value:

- 2 A resource browse or a conflicting browse is already in progress.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

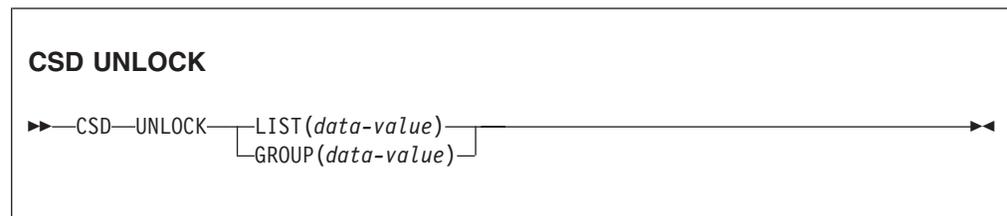
NOTFND

RESP2 value:

- 2 The specified group cannot be found.

CSD UNLOCK

Remove the lock from a group or list of definitions.



Conditions: CSDERR, DUPRES, INVREQ, LOCKED, NOTAUTH

Description

The **UNLOCK** command removes from a group or list a lock previously added by the **LOCK** command.

A syncpoint is implicit in **CSD UNLOCK** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the command is successful, and rolled back if not.

Options

GROUP (*data-value*)

Specifies the 8-character name of the group to be unlocked.

LIST (*data-value*)

Specifies the 8-character name of the list to be unlocked.

Conditions

CSDERR

RESP2 values:

- 1 The CSD cannot be read.
- 2 The CSD is read only.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

DUPRES

RESP2 values:

- 2 The name specified in **GROUP** exists in the CSD as a list.
- 3 The name specified in **LIST** exists in the CSD as a group.

INVREQ

RESP2 values:

- 2 The **GROUP** option contains one or more characters that are not valid.
- 3 The **LIST** option contains one or more characters that are not valid.
- 200 The command was run in a program defined with an **EXECUTIONSET** value of **DPLSUBSET** or in a program invoked from a remote system by a distributed program link without the **SYNCONRETURN** option.

LOCKED

RESP2 values:

- 1 The specified group or list is already locked to another user.
- 2 The group or list is IBM-protected.

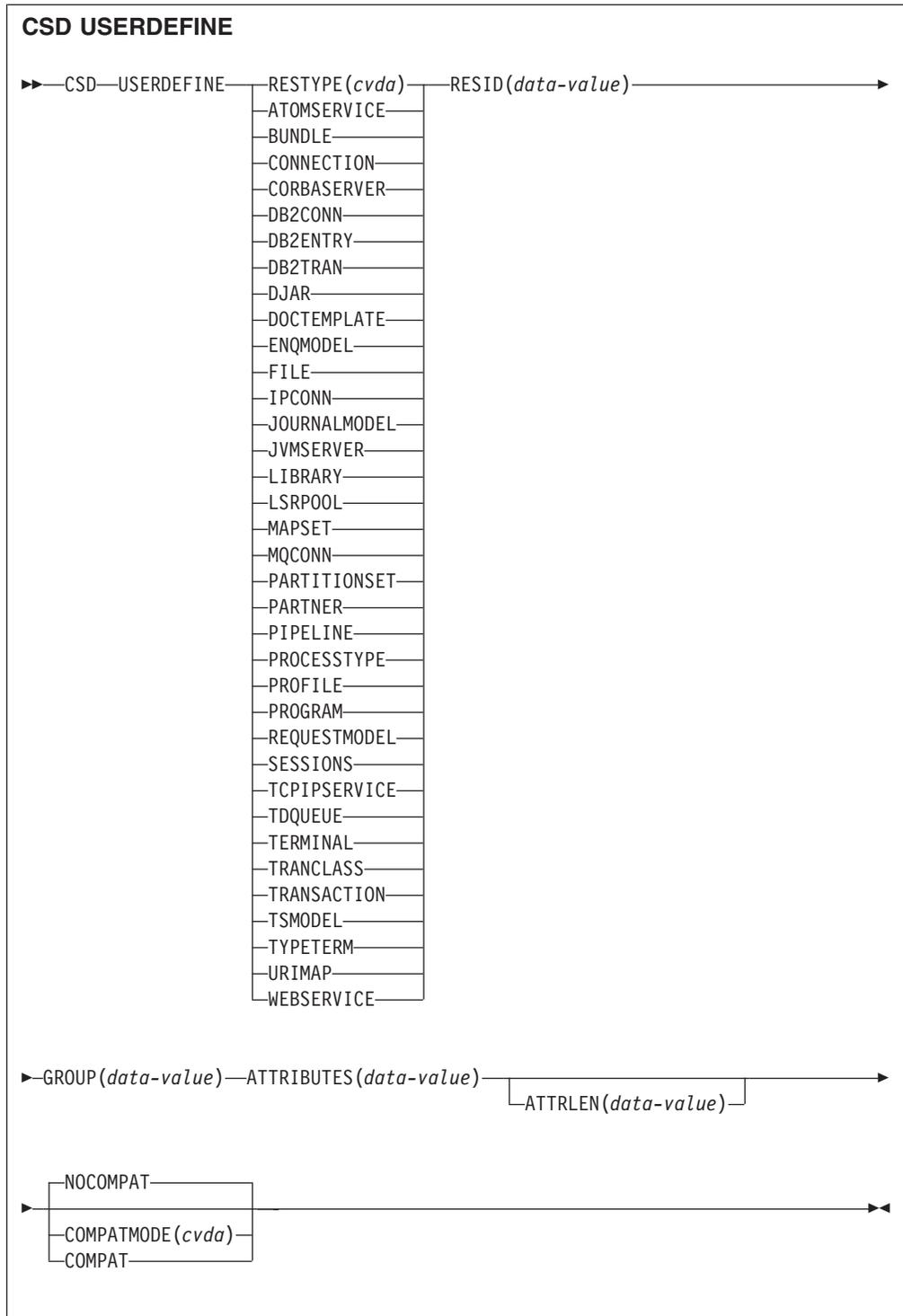
NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

CSD USERDEFINE

Create a new resource definition with user-specified default values in the CSD.



Conditions: CSDERR, DUPRES, INVREQ, LENGERR, LOCKED, NOTAUTH, NOTFND

Description

USERDEFINE is an alternative to the **DEFINE** command. Instead of using CICS-supplied default values, **USERDEFINE** uses your own defaults. Otherwise, it operates in exactly the same way as **DEFINE**.

To set up your own defaults, use **DEFINE** to create a dummy resource definition named **USER** in a group named **USERDEF**. Each dummy resource definition must be complete, for example, a transaction definition must name a program definition, even though you always supply a program name when you **USERDEFINE** a transaction. You need not install the dummy resource definitions before using **USERDEFINE**.

Create a dummy resource definition for each type of resource for which you want to set default values. Each of them is named **USER**, but because they are definitions of different resources, they are unique.

A syncpoint is implicit in **CSD USERDEFINE** processing, except when an exception condition is detected early in processing the command. Uncommitted changes to recoverable resources made up to that point in the task are committed if the command is successful, and rolled back if not.

Options

ATTRIBUTES (*data-value*)

Specifies the attributes of the new resource. Code the list of attributes as a single character string.

See the *CICS Resource Definition Guide* for details about specific attributes.

ATTRLEN (*data-value*)

Specifies the length in bytes of the character string supplied in the **ATTRIBUTES** option, as a fullword binary value.

COMPATMODE (*cvda*)

Specifies whether obsolete attributes are allowed in the **ATTRIBUTES** string for this command. Specify one of the following **CVDA** values:

COMPAT

Obsolete resource attributes are allowed in the **ATTRIBUTES** string for this command.

NOCOMPAT

Obsolete resource attributes are not allowed in the **ATTRIBUTES** string for this command.

The default is **NOCOMPAT**.

GROUP (*data-value*)

Specifies the 8-character name of the group to which the resource definition is to belong.

RESID (*data-value*)

Specifies the 8-character name of the resource to be altered. Resource names such as **TRANSACTION** that are only four characters must be padded with four blanks and passed in an 8-character field.

RESTYPE (*cvda*)

Specifies the type of resource to be defined. **CVDA** values are the resource type names.

Conditions

CSDERR

RESP2 values:

1 The CSD cannot be read.

- 2 The CSD is read only.
- 3 The CSD is full.
- 4 The CSD is being used by another CICS system and is not configured for sharing.
- 5 Insufficient VSAM strings (**CSDSTRNO** system initialization parameter value) are available to run the **EXEC CICS CSD** command.

DUPRES

RESP2 values:

- 1 A resource of this name and type already exists in the specified group.
- 2 The specified group did not exist but because a list of the same name is already present in the CSD, the group could not be created.

INVREQ

RESP2 values:

- 1 The resource type specified for RESTYPE is not valid.
- 2 The GROUP option contains one or more characters that are not valid.
- 11 The value specified for COMPATMODE is not valid.
- 200 The command was run in a program defined with an EXECUTIONSET value of DPLSUBSE, or in a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.
- n The ATTRIBUTES string contains a syntax error or RESID contains a character that is not valid.

LENGERR

RESP2 value:

- 1 The length specified in ATTRLEN is negative.

LOCKED

RESP2 values:

- 1 The specified group is locked to another user.
- 2 The group is IBM-protected.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

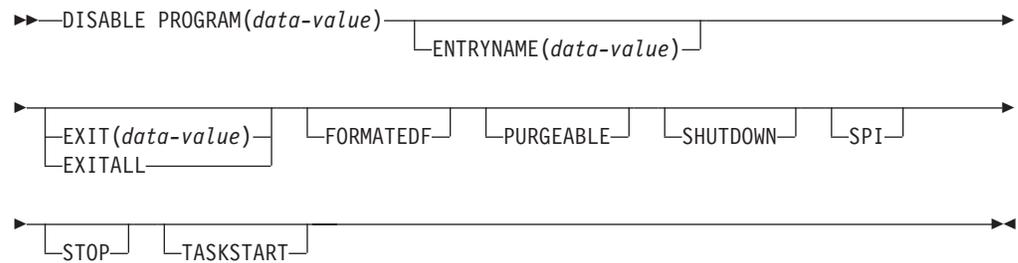
RESP2 values:

- 5 The group USERDEF cannot be found.
- 6 A resource of the required type with name USER cannot be found in group USERDEF.

DISABLE PROGRAM

Terminate or otherwise modify the invocation of a user exit.

DISABLE PROGRAM



Conditions: INVEXITREQ, NOTAUTH

Description

The **DISABLE PROGRAM** command changes the status of a global or task-related user exit, reversing the effects of corresponding options in an **ENABLE PROGRAM** command.

You use it to:

- Remove points at which a particular exit is invoked
- Make the exit unavailable for execution (without removing its status as an exit)
- Delete its definition as an exit entirely.

Options on the **DISABLE PROGRAM** command correspond to those on the **ENABLE** command:

- **ENTRYNAME** and **PROGRAM** identify the exit to be disabled, and you must use exactly the same combination of values that you did in the **ENABLE** command that defined the exit.
- **EXIT**, **FORMATEDF**, **SHUTDOWN**, and **TASKSTART** reverse the effect of the same-named options on **ENABLE PROGRAM**; that is, they turn off invocation of the exit at the points specified.
- **STOP** reverses the effect of **START**, making the exit unavailable for execution.
- **EXITALL** deletes the definition entirely, reversing the effect of the **ENABLE PROGRAM** that created the exit. Work areas and the load module associated with the exit may be deleted as well.

For programming information about CICS exits, see Global user exit programs and Task-related user exit programs in the Customization Guide; you should also see the general discussion of commands that modify exits in “Exit-related commands” on page 28.

Note: One or more of **STOP** and (**EXIT** or **EXITALL**) is required for a global user exit, and one or more of **STOP**, **EXITALL**, **TASKSTART**, **SHUTDOWN**, and **FORMATEDF** is required for a task-related user exit.

Options

ENTRYNAME (*data-value*)

Specifies the name of the global or task-related user exit whose status is to be changed. If you omit **ENTRYNAME**, CICS assumes that the name of the exit is the same as the load module name given in the **PROGRAM** option. Therefore,

you must use the same combination of ENTRYNAME and PROGRAM values on DISABLE commands as was specified on the initial ENABLE command that defined the exit.

EXIT(*data-value*) (global user exits only)

Specifies the name of the global user exit point from which this exit program is to be dissociated. It causes CICS to stop invoking the exit at this point but does not, of itself, cause CICS to delete the associated load module from virtual storage, even if it is no longer being used at any exit points. Exit point names are eight characters long.

EXITALL

Causes CICS to discard the definition of the exit. For a global user exit, EXITALL dissociates the exit from *all* of the exit points from which it currently is invoked. If possible, the associated load module is deleted from virtual storage.

For a task-related user exit, the associated load module is deleted from virtual storage if it is not in use by another exit and if the ENTRY option was not specified in the ENABLE command that defined the exit. If the exit owns a global work area, the work area is released as soon as no other exits are sharing it.

EXITALL implies STOP, so the exit becomes unavailable for execution. For a task-related user exit, you must avoid requesting this function until all tasks that have used the exit have ended; the results of EXITALL before that point are unpredictable. This means that for start-of-task, end-of-task and shutdown calls, when all task related user exits would be called and a **DISABLE EXITALL** command is issued from the current TRUE for itself then the number of TRUES called is unpredictable. If the need arises for the exit to be refreshed then the TRUE should disable itself using the STOP option and invoke a separate task to issue the DISABLE with the EXITALL option.

FORMATEDF (task-related user exits only)

Indicates that the exit should not be invoked to format EDF screens. You can reinstate invocation at EDF points with an ENABLE command specifying FORMATEDF.

PROGRAM(*data-value*)

Specifies the 8-character name of the **load module** that contains the entry point for the exit. This name is also used as the name of the exit when ENTRYNAME is not specified; see the ENTRYNAME option.

PURGEABLE (task-related user exits only)

Removes the ability to be purged from CICS waits while active in the task-related user exit. You can turn this on with an ENABLE command specifying PURGEABLE.

SHUTDOWN (task-related user exits only)

Indicates that the exit should not be invoked at CICS shutdown. You can reinstate invocation at shutdown with an ENABLE command specifying SHUTDOWN.

SPI (task-related user exits only)

Specifies that the task-related user exit is no longer to be invoked if an **INQUIRE EXITPROGRAM** command specifies the CONNECTST or QUALIFIER option, or both.

STOP

Specifies that the exit is to be made unavailable for execution, but is to remain enabled (defined as an exit). You can make the exit available for execution again with an ENABLE command specifying START.

When a stopped task-related user exit gets invoked, the invoking code gets an AEY9 abend code. There is no corresponding error for global user exits, however, because CICS invokes only those exits associated with an exit point which are also available for execution (not stopped).

TASKSTART (task-related user exits only)

Indicates that the exit should not be invoked at the start and end of each task. You can reinstate these invocations with an ENABLE command specifying TASKSTART.

Conditions

INVEXITREQ

The INVEXITREQ condition of the DISABLE command is indicated by X'80' in the first byte of EIBRCODE. The exact cause of the error can be determined by examining the second and third bytes of EIBRCODE, which can have the values shown in the following list.

X'808000'

The load module named on the PROGRAM parameter has not been defined to CICS, or the load module is not in the load library, or the load module has been disabled.

X'804000'

The value of EXIT is not a valid exit point.

X'800200'

The exit identified by the PROGRAM value is not defined as an exit.

X'800100'

The exit identified by ENTRYNAME is not defined as an exit.

X'800080'

The exit is currently invoked by another task (see note).

Note: The INVEXITREQ condition with X'0080' in the second and third bytes can occur:

- If you issue the DISABLE request while a task using the exit has been suspended temporarily because of a request for a CICS service within the exit. The normal action for this condition is to retry the DISABLE request.
- When a DISABLE request with EXITALL or EXIT has been specified, but the exit has already terminated abnormally. In this case, the use count of the associated load module remains greater than zero. The exit cannot be dissociated from any exit point, and the load module cannot be deleted from virtual storage. The exit can, however, be made unavailable for execution by issuing a DISABLE STOP command.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

Examples

1. The following example makes exit EP2 non-executable. It does not dissociate it from the exit points with which it is associated, however, or delete its definition as an exit. It can be made available again by issuing an ENABLE PROGRAM('EP2') START command.

```
EXEC CICS DISABLE PROGRAM('EP2') STOP
```

2. The following example stops global user exit ZX from being invoked at exit point XTDREQ. ZX is still defined, however, and if it is associated with other exit points, it is still invoked at them.

```
EXEC CICS DISABLE ENTRYNAME ('ZX') PROGRAM('EP3')  
EXIT('XTDREQ')
```

3. The following example dissociates EP3 from all points at which invocation was requested (exit points, in the case of a global user exit; task start, shutdown, and so on, in the case of a task-related user exit), and discards the definition of the exit. If the load module EP3 is not in use, it is deleted.

```
EXEC CICS DISABLE PROGRAM('EP3') EXITALL
```

DISCARD ATOMSERVICE

Remove an ATOMSERVICE resource definition from the system.

DISCARD ATOMSERVICE

```
►►—DISCARD ATOMSERVICE(data-value)—◄◄
```

Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

The DISCARD ATOMSERVICE command removes a disabled ATOMSERVICE definition from the system.

Options

ATOMSERVICE (*data-value*)

Specifies the 8-character name of the ATOMSERVICE definition that is to be removed.

Conditions

INVREQ

RESP2 values are:

- 4 The ATOMSERVICE is not disabled.
- 200 The command ran in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

NOTFND

RESP2 values:

- 3 The ATOMSERVICE cannot be found.

DISCARD AUTINSTMODEL

Remove a terminal autoinstall model definition.

DISCARD AUTINSTMODEL

►►—DISCARD AUTINSTMODEL(*data-value*)—►►

Conditions: INVREQ, MODELIDERR, NOTAUTH

Description

The DISCARD AUTINSTMODEL command makes a TERMINAL definition in the local CICS system ineligible for use as a model for automatic installation of terminals. The TERMINAL definition is not discarded or otherwise modified; it is only removed from the list of autoinstall models available. (Use the DISCARD TERMINAL command if you want to remove the definition of the terminal.)

See “Discarding resource definitions” on page 27 for general information about discards.

Options

AUTINSTMODEL (*data-value*)

specifies the 8-character name of the autoinstall model that is to be removed. This is the name specified in the AUTINSTNAME option of the TERMINAL definition that defines the model, or the name of the terminal if AUTINSTNAME was not specified.

Models whose names begin with the letters DFH are assumed to be CICS-supplied models and cannot be discarded.

Conditions

INVREQ

RESP2 values:

- 2 The model you requested is currently in use.
3 The model cannot be discarded because its name begins with DFH.

MODELIDERR

RESP2 values:

- 1 The model cannot be found.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

DISCARD BUNDLE

Remove a BUNDLE resource definition.

DISCARD BUNDLE

►►—DISCARD BUNDLE(*data-value*)—◄◄

Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the **DISCARD BUNDLE** command to remove a BUNDLE resource from your CICS region. All the resources that were dynamically created must be disabled or unusable before the BUNDLE resource can be discarded.

Options

BUNDLE(*data-value*)

Specifies the 8-character name of the BUNDLE definition that is to be discarded.

Conditions

INVREQ

RESP2 values:

- 5 The BUNDLE is not disabled.
- 6 The bundle is in an invalid state. You must disable the bundle before it can be discarded.
- 7 CICS failed to link to the registered bundle callback program.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

NOTFND

RESP2 values:

- 3 The BUNDLE cannot be found.

DISCARD CONNECTION

Remove a CONNECTION definition.

DISCARD CONNECTION

►►—DISCARD CONNECTION(*data-value*)—◄◄

Conditions: INVREQ, NOTAUTH, SYSIDERR

Description

The DISCARD CONNECTION command removes a CONNECTION definition from the local CICS system. When a connection is removed, all of the associated sessions also are removed.

For deletion to be successful:

- The connection can have no active sessions if it is remote (that is, has a REMOTESYSTEM value other than the name of the local region), and must be in OUTSERVICE status if it is not remote.
- The interregion communications facility must be closed if the connection is an MRO connection. (You can use the SET IRC CLOSED command to close it.)
- If the connection is an APPC connection and the local CICS system is a member of a z/OS Communications Server generic resource group, there can be no deferred work pending. Deferred work occurs when a failure causes a unit of work which has used a session on the connection at SYNCLEVEL 2 to be “shunted” (held for later disposition, because recovery action is required before disposition can be completed).

Other types of connection *can* be discarded, even if there is recovery work outstanding for them. However, it is recommended that you do not discard them if there is. You can use the INQUIRE CONNECTION RECOVSTATUS command to check.

- There can be no indirect CONNECTION definitions pointing to the connection to be discarded.

Note: In unusual circumstances, the discard of an LU6.1 connection can fail, even when it is out-of-service, if some of its sessions are still in-service. If this happens, set the connection status to INSERVICE, then OUTSERVICE, and then reissue the DISCARD command.

CICS completes successful DISCARD CONNECTION processing with an implicit syncpoint on behalf of the issuing task, committing changes to recoverable resources made up to that point in the task. If the discard processing fails, CICS raises the INVREQ exception condition with a RESP2 value of 27, and does a SYNCPOINT ROLLBACK instead, rolling back changes to recoverable resources. For all other exception conditions, however, discard processing is not attempted and neither SYNCPOINT nor SYNCPOINT ROLLBACK is issued.

See “Discarding resource definitions” on page 27 for general information about DISCARD commands.

Options

CONNECTION(*data-value*)

specifies the 4-character identifier of the CONNECTION definition to be discarded.

Conditions

INVREQ

RESP2 values:

- 24 The connection is remote and is in use locally.
- 25 The connection is local and is not out-of-service.

- 26 Recovery information is outstanding for the connection which must be resolved before discard is allowed.
- 27 Discard processing failed.
- 28 Indirect connections point to the connection.
- 29 The connection is an MRO connection and IRC is not closed.
- 38 Discard of this connection is already in progress.
- 39 The CONNECTION definition is currently in use.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SYSIDERR

RESP2 values:

- 9 The connection cannot be found.

DISCARD CORBASERVER

Remove the definition of a CorbaServer from the system, together with any associated deployed JAR files and beans.

DISCARD CORBASERVER

►►—DISCARD CORBASERVER(*data-value*)—◄◄

Conditions: INVREQ,NOTAUTH, NOTFND

Description

The DISCARD CORBASERVER command allows you to remove the definition of a CorbaServer from the system, together with any associated deployed JAR files and beans.

If any of the beans installed in the named CorbaServer are in use when the DISCARD command executes, the state of the CorbaServer turns to DISCARDING. No new work is accepted. When all outstanding work has completed, the CORBASERVER definition is removed.

Copies of deployed JAR files are deleted from the CorbaServer's shelf, together with any other files on the shelf that belong to the CorbaServer. The CorbaServer's shelf directory is deleted.

Conditions

INVREQ

RESP2 values:

- 22 Discard is in progress for this deployed Corbaserver.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

1 The named CorbaServer was not found.

DISCARD DB2CONN

Remove a DB2CONN definition.

DISCARD DB2CONN

►►—DISCARD DB2CONN(*data-value*)—————►►

Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

The DISCARD DB2CONN command removes the definition of a DB2CONN from the local CICS system; that is, it revokes the earlier installation of a DB2CONN resource definition.

A DB2CONN can only be discarded when the CICS DB2 interface is not active.

Remember: A discard of a DB2CONN also implicitly discards all DB2ENTRYs and DB2TRANs currently installed.

Options

None

Conditions**INVREQ**

RESP2 values:

2 The CICS DB2 interface is active.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

1 A DB2CONN cannot be found.

DISCARD DB2ENTRY

Remove a DB2ENTRY definition.

DISCARD DB2ENTRY

▶▶—DISCARD DB2ENTRY(*data-value*)—————▶▶

Conditions: NOTFND, INVREQ, NOTAUTH

This command is threadsafe.

Description

The DISCARD DB2ENTRY command removes the definition of a DB2ENTRY from the local CICS system, so that the system no longer has access to the DB2ENTRY; that is, it revokes the earlier installation of a DB2ENTRY resource definition of the same name.

A DB2ENTRY must be disabled for its definition to be discarded.

Options

DB2ENTRY(*data-value*)

specifies the 8-character name of the DB2ENTRY that is to be removed.

Conditions

NOTFND

RESP2 values:

- 1 The DB2ENTRY cannot be found.

INVREQ

RESP2 values:

- 2 The DB2ENTRY is currently in use.
- 3 The DB2ENTRY is not disabled.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

DISCARD DB2TRAN

Remove a DB2TRAN definition.

DISCARD DB2TRAN

▶▶—DISCARD DB2TRAN(*data-value*)—————▶▶

Conditions: NOTFND, NOTAUTH

This command is threadsafe.

Description

The DISCARD DB2TRAN command removes the definition of a DB2TRAN from the local CICS system, so that the transaction id specified in the DB2TRAN no longer uses the named DB2ENTRY; that is, it revokes the earlier installation of a DB2TRAN resource definition of the same name.

A DB2TRAN can be discarded at any time.

Options

DB2TRAN(*data-value*)

specifies the 8-character name of the DB2TRAN that is to be removed.

Conditions

NOTFND

RESP2 values:

- 1 The DB2TRAN cannot be found.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access the DB2ENTRY referenced by this DB2TRAN in the way required by this command.

DISCARD DJAR

Remove the definition of a specified deployed JAR file from the system, together with any associated beans.

DISCARD DJAR

►►—DISCARD DJAR(*data-value*)—◄◄

Conditions: INVREQ,NOTAUTH, NOTFND

Description

The DISCARD DJAR command removes the definition of a deployed JAR file from the system, together with the shelf copy of the deployed JAR file and the beans it contains. The current set of JVM instances are marked for termination, so that each JVM instance will not be reused after it terminates. New JVM instances load all their classes and do not use old versions of classes whose bytecodes have been replaced.

The copy of the deployed JAR file that was created when the deployed JAR file was installed is deleted from the CorbaServer's shelf.

Passivated stateful session bean instances are not deleted by this operation because, in a workload-managed CICS EJB server, they could be activated by other AORs .

References to the homes of enterprise beans in the deployed JAR file are not retracted from the namespace because other AORs might still have the homes installed. However, before discarding a deployed JAR file from the last AOR in which it is installed, you might want to issue a PERFORM DJAR RETRACT command to retract references to home interfaces.

Options

DJAR(*data-value*)

specifies the 1–32 character name of the deployed JAR file to be discarded.

Conditions

INVREQ

RESP2 values:

22 Delete is in progress for this deployed JAR file.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

1 The named deployed JAR file was not found.

DISCARD DOCTEMPLATE

Remove a document template.

DISCARD DOCTEMPLATE

▶▶—DISCARD DOCTEMPLATE(*data-value*)—————▶▶

Conditions: NOTAUTH, NOTFND

This command is threadsafe.

Description

The DISCARD DOCTEMPLATE command removes a document template definition from the local CICS system, so that the system no longer has access to the resource (that is, it revokes the earlier installation of an DOCTEMPLATE definition of the same name).

See “Discarding resource definitions” on page 27 for general information about discards.

Options

DOCTEMPLATE(*data-value*)

specifies the 8-character name of the DOCTEMPLATE definition that you want to remove.

Conditions

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this DOCTEMPLATE resource definition in the way required by this command.

NOTFND

RESP2 values:

- 1 The specified DOCTEMPLATE resource definition is not installed on this system.

DISCARD ENQMODEL

Remove an ENQMODEL resource definition.

DISCARD ENQMODEL

►►—DISCARD ENQMODEL(*data-value*)—————◄◄

Conditions: NOTAUTH, NOTFND

Description

The DISCARD ENQMODEL command removes the definition of an ENQ model from the local CICS system. When discard is issued, the ENQMODEL is put into the WAITING state until there are no enqueues in the local system which match the ENQNAME pattern. It is then removed from the local system, so that the system no longer has access to the ENQMODEL; that is, it revokes the earlier installation of an ENQMODEL resource definition of the same name.

Adding or removing a definition does not affect enqueues already held, only ENQ commands issued after the definition is added or removed are affected.

See “Discarding resource definitions” on page 27 for general information about discards.

Options

ENQMODEL(*data-value*)

specifies the 8-character identifier of the ENQ model that is to be discarded.

Conditions

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

- 1 The specified ENQMODEL is not installed on this system.

DISCARD FILE

Remove a FILE definition.

DISCARD FILE

▶—DISCARD FILE(*data-value*)—————▶

Conditions: FILENOTFOUND, INVREQ, NOTAUTH

Description

The DISCARD FILE command removes the definition of a file from the local CICS system, so that the system no longer has access to the file; that is, it revokes the earlier installation of a FILE resource definition of the same name.

A file must be closed and disabled for its definition to be discarded. In addition, if the file is recoverable, it cannot be discarded until all retained locks on it are released. A lock is retained when a failure causes a unit of work which has modified the file to be “shunted” (held for later disposition, because recovery action is required before disposition can be completed).

See “Discarding resource definitions” on page 27 for general information about discards.

Options

FILE(*data-value*)

specifies the 8-character name of the file that is to be removed.

You cannot remove the definition of a file whose name begins with the letters DFH, because such files are reserved for CICS.

Conditions

FILENOTFOUND

RESP2 values:

18 The file cannot be found.

INVREQ

RESP2 values:

2 The file is not closed.

3 The file is not disabled.

25 The FILE definition is currently in use.

26 The file cannot be discarded because its name begins with DFH.

43 The file cannot be discarded because it has deferred work outstanding, for which there are retained locks.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

DISCARD IPCONN

Remove an IPCONN definition.

DISCARD IPCONN

►►—DISCARD IPCONN(*data-value*)—◄◄

Conditions: INVREQ, NOTAUTH, SYSIDERR

Description

The DISCARD IPCONN command removes an IPCONN definition from the local CICS system.

You cannot discard an IPCONN unless it is in OUTSERVICE status.

See “Discarding resource definitions” on page 27 for general information about DISCARD commands.

Options

IPCONN(*data-value*)

specifies the 8-character name of the IPCONN definition to be discarded.

Conditions

INVREQ

RESP2 values:

- 5 The IPCONN is in service.
- 9 The IPCONN does not exist.
- 27 Discard processing failed.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SYSIDERR

RESP2 values:

- 9 The IPCONN name was not found.

DISCARD JOURNALMODEL

Remove a journal model definition.

DISCARD JOURNALMODEL

►►—DISCARD JOURNALMODEL(*data-value*)—◄◄

Conditions: NOTAUTH, NOTFND

Description

The DISCARD JOURNALMODEL command makes a JOURNALMODEL definition ineligible as a model for defining journals in local CICS system. The JOURNALMODEL definition itself is not discarded or otherwise modified, nor is there any effect on existing journals defined using the model. These journals continue to use their existing definitions unless they are discarded using a DISCARD JOURNALNAME command.

See “Discarding resource definitions” on page 27 for general information about discards.

Options

JOURNALMODEL (*data-value*)

specifies the 8-character name of the journal model that you want to remove.

Conditions

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

1 The journal model name was not found.

DISCARD JOURNALNAME

Remove a journal name from the journal names table.

DISCARD JOURNALNAME

►►—DISCARD JOURNALNAME(*data-value*)—◄◄

Conditions: INVREQ, JIDERR, NOTAUTH

Description

The DISCARD JOURNALNAME command removes a journal definition from the local CICS system, so that the next time the journal definition is used, it is recreated based on the current set of JOURNALMODEL definitions. Thus you can use it in conjunction with DISCARD and CREATE JOURNALMODEL commands to change the definition of a particular journal.

The command takes effect immediately for user journals, including the “log of logs” journal, and for terminal control autojournals. On the next reference to the journal following the DISCARD, a new journal definition is created using attributes from the JOURNALMODEL definition that matches best at that time.

For forward recovery and auto-journaling journals, however, the journal definition is used only when one of the files using the journal is opened. Hence the command has no effect on forward-recovery logging or auto-journaling operations

for VSAM files that are open and using the journal at the time of the DISCARD. They continue to use the log stream referenced by the existing journal until the files are closed, and are not affected by the DISCARD unless the file is subsequently reopened. In addition, if the logstream identifier is present in the VSAM catalog definition for a file, as it must be for an RLS file and may be for others, the catalog value overrides the JOURNALMODEL value.

Neither component of the CICS system log, DFHLOG or DFHSHUNT, is eligible for discard.

See “Discarding resource definitions” on page 27 for general information about discards.

Options

JOURNALNAME (*data-value*)

specifies the 8-character name of the journal that you want to remove.

Note: To discard a journal defined with a numeric identifier specify the journal name as DFHJ*nn*, where *nn* is the two-digit journal number, in the range 01–99.

Conditions

INVREQ

RESP2 values:

3 The journal specified cannot be discarded.

JIDERR

RESP2 values:

1 The journal cannot be found.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

DISCARD JVMSERVER

Remove a JVMSERVER resource definition.

DISCARD JVMSERVER

►►—DISCARD JVMSERVER(*data-value*)—◄◄

Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the DISCARD JVMSERVER command to remove a JVMSERVER resource from your CICS region. The JVMSERVER resource must be disabled before it can be discarded.

Options

JVMSERVER(*data-value*)

Specifies the 8-character name of the JVMSERVER resource definition that is to be discarded.

Conditions

INVREQ

RESP2 value:

5 The JVMSERVER is not disabled.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

NOTFND

RESP2 value:

3 The JVMSERVER cannot be found.

DISCARD LIBRARY

Remove a specified LIBRARY from the running CICS system.

DISCARD LIBRARY

►—DISCARD LIBRARY(*data-value*)—◄

Conditions: INVREQ, NOTAUTH, NOTFND

Description

The DISCARD LIBRARY command removes the definition of a LIBRARY from the local CICS system, so that the system no longer has access to the LIBRARY, that is, it revokes the earlier installation of a LIBRARY resource definition of the same name. A LIBRARY must be disabled for its definition to be discarded.

See “Discarding resource definitions” on page 27 for general information about discards.

Options

LIBRARY(*data-value*)

specifies the 8-character name of the LIBRARY that is to be removed.

Conditions

INVREQ

RESP2 values:

- 3 The LIBRARY is not disabled.
- 6 The LIBRARY name is DFHRPL, and the static DFHRPL cannot be discarded.
- 7 A failure was encountered on the second attempt at deconcatenating the LIBRARY data sets.
- 8 A failure occurred on the second attempt at deallocating the LIBRARY data sets.
- 9 A failure occurred while attempting to delete LIBRARY control structures.
- 10 A failure occurred on the second attempt at closing the LIBRARY.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

- 1 The named LIBRARY cannot be found.

DISCARD MQCONN

Remove an MQCONN resource definition. Any implicit MQINI resource definition is also discarded.

DISCARD MQCONN

▶▶—DISCARD MQCONN—▶▶

Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

The DISCARD MQCONN command removes an installed MQCONN resource definition from the local CICS system. Only one MQCONN resource definition can be installed in a CICS system at a time, so no name or identifier is required on this command.

An MQCONN resource definition can be discarded only when CICS is not connected to WebSphere MQ.

If the installed MQCONN resource definition had an INITQNAME attribute, CICS installed an implicit MQINI resource definition to represent the default initiation queue. When you discard the MQCONN resource definition, CICS discards any associated MQINI resource definition.

Options

None

Conditions

INVREQ

RESP2 values:

- 2 CICS is connected to WebSphere MQ.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

- 1 An MQCONN resource definition cannot be found.

DISCARD PARTNER

Remove a PARTNER definition.

DISCARD PARTNER

▶▶—DISCARD PARTNER(*data-value*)—▶▶

Conditions: INVREQ, NOTAUTH, PARTNERIDERR

Description

The DISCARD PARTNER command removes the definition of a partner from the local CICS system, so that the system no longer has access to the partner; that is, it revokes the earlier installation of a PARTNER resource definition of the same name.

See “Discarding resource definitions” on page 27 for general information about discards.

Options

PARTNER(*data-value*)

specifies the 8-character name of the partner that is to be removed.

Partners whose names begin with the letters DFH are assumed to be CICS-defined partners and cannot be discarded.

Conditions

INVREQ

RESP2 values:

- 2 The PARTNER definition is currently in use.
- 3 The partner cannot be discarded because its name begins with DFH.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

PARTNERIDERR

RESP2 values:

- 1 The partner cannot be found.
- 5 The Partner Resource Manager (PRM) is not active, because it failed to initialize during CICS initialization.

DISCARD PIPELINE

Remove a PIPELINE definition.

DISCARD PIPELINE

►►—DISCARD PIPELINE(*data-value*)—◄◄

Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the DISCARD PIPELINE to remove an PIPELINE from your CICS region. The PIPELINE must be disabled before it can be discarded.

See “Discarding resource definitions” on page 27 for general information about discards.

Options

PIPELINE(*data-value*)

specifies the 8-character name of the PIPELINE whose definition is to be discarded.

Conditions

INVREQ

RESP2 values:

- 8 The PIPELINE cannot be discarded because it is not disabled.
- 22 Delete is in progress for this PIPELINE.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

- 1 The PIPELINE cannot be found

DISCARD PROCESSTYPE

Remove a PROCESSTYPE definition.

DISCARD PROCESSTYPE

►►—DISCARD PROCESSTYPE(*data-value*)—◄◄

Conditions: INVREQ, NOTAUTH, PROCESSERR

Description

The DISCARD PROCESSTYPE command removes a CICS business transaction services (BTS) PROCESSTYPE definition from the local CICS region.

Note:

1. Only disabled process-types can be discarded.
2. If you are using BTS in a single CICS region, you can use the DISCARD PROCESSTYPE command to remove process-types. However, if you are using BTS in a sysplex, it is strongly recommended that you use CPSM to remove them. If you don't use CPSM, problems could arise if Scheduler Services routes to this region work that requires a discarded definition.

See “Discarding resource definitions” on page 27 for general information about discards.

Options

PROCESSTYPE(*data-value*)

specifies the 8-character name of the PROCESSTYPE that you want to remove.

Conditions

INVREQ

RESP2 values:

- 2 The process-type named in the PROCESSTYPE option is not disabled.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

PROCESSERR

RESP2 values:

- 1 The process-type named in the PROCESSTYPE option is not defined in the process-type table (PTT).

DISCARD PROFILE

Remove a PROFILE definition.

DISCARD PROFILE

►►—DISCARD PROFILE(*data-value*)—◄◄

Conditions: INVREQ, NOTAUTH, PROFILEIDERR

Description

The DISCARD PROFILE command removes the definition of a profile from the local CICS system, so that the system no longer has access to the profile; that is, it revokes the earlier installation of a PROFILE resource definition of the same name. You cannot discard a profile while any installed TRANSACTION definitions point to it.

See “Discarding resource definitions” on page 27 for general information about discards.

Options

PROFILE(*data-value*)

specifies the 8-character name of the profile that is to be removed.

Profiles whose names begin with the letters DFH are assumed to be CICS-supplied profiles and cannot be discarded.

Conditions

INVREQ

RESP2 values:

- 2 The PROFILE definition is currently in use.
- 3 A TRANSACTION definition points to the profile.
- 4 The profile cannot be discarded because its name begins with DFH.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

PROFILEIDERR

RESP2 values:

- 1 The profile cannot be found.

DISCARD PROGRAM

Remove the definition of a program, map set, or partition set.

DISCARD PROGRAM

►►—DISCARD PROGRAM(*data-value*)—◄◄

Conditions: INVREQ, NOTAUTH, PGMIDERR

Description

The DISCARD PROGRAM command removes the definition of a program, map set, or partition set (a load module resource) from the local CICS system, so that the system no longer has access to the resource; that is, it revokes the earlier installation of a PROGRAM, MAPSET, or PARTITIONSET definition of the same name.

You cannot discard a module that is being executed or otherwise used by a task. Definitions supplied by CICS (modules whose name begin with DFH) and modules defined as user-replaceable (such as autoinstall programs) also are ineligible.

See “Discarding resource definitions” on page 27 for general information about discards.

Options

PROGRAM(*data-value*)

specifies the 8-character name of the program, map set, or partition set that is to be removed.

Conditions

INVREQ

RESP2 values:

- 1 The resource cannot be discarded because its name begins with DFH.
- 11 The resource definition is currently in use.
- 15 The resource cannot be discarded because it is a user-replaceable module.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

PGMIDERR

RESP2 values:

- 7 The resource definition cannot be found.

DISCARD REQUESTMODEL

Remove a request model definition.

DISCARD REQUESTMODEL

►►—DISCARD REQUESTMODEL(*data-value*)—►►

Conditions: NOTAUTH, NOTFND

Description

The DISCARD REQUESTMODEL command makes a REQUESTMODEL definition ineligible as a model for defining requests in the local CICS system. The REQUESTMODEL definition itself is not deleted or otherwise modified.

See “Discarding resource definitions” on page 27 for general information about discards.

Options

REQUESTMODEL(*data-value*)

specifies the 8-character name of the request model that you want to remove.

Conditions

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

1 The request model name was not found.

DISCARD TCPIP SERVICE

Remove a TCPIP SERVICE definition.

DISCARD TCPIP SERVICE

►►—DISCARD TCPIP SERVICE(*data-value*)—◀◀

Conditions: INVREQ, NOTAUTH, NOTFND

Description

The DISCARD TCPIP SERVICE command removes a TCPIP SERVICE definition from the local CICS system.

You cannot discard a TCPIP SERVICE unless it is in CLOSED status, showing that it is not in use.

See “Discarding resource definitions” on page 27 for general information about discards.

Options

TCPIP SERVICE(*data-value*)

specifies the 8-character name of the TCPIP SERVICE that you want to remove.

Conditions

INVREQ

RESP2 values:

9 The TCPIP SERVICE is still open.

16 The TCPIP SERVICE cannot be discarded because it is referred to by an installed CORBASERVER definition.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

- 3 The TCPIPSERVICE name was not found.

DISCARD TDQUEUE

Remove a transient data queue definition.

DISCARD TDQUEUE

►►—DISCARD TDQUEUE(*data-value*)—————►►

Conditions: INVREQ, NOTAUTH, QIDERR

Description

The DISCARD TDQUEUE command removes the definition of a transient data queue from the local CICS system.

A queue must be disabled before it can be discarded, and an extrapartition queue must be closed as well. See “Discarding resource definitions” on page 27 for rules governing disabling of queues. Queues required by CICS (those whose names begin with the letter C) cannot be discarded.

When an intrapartition queue is discarded, an implicit DELETEDQ command is executed to empty the queue and release space in the data set associated with it. If the queue is defined as logically recoverable, an implicit SYNCPOINT command follows the DELETEDQ. The SYNCPOINT commits all changes to recoverable resources made up to that point in the task that issued the DISCARD TDQUEUE command. However, deletion and syncpoint take place only if the command completes successfully, without raising any exception conditions.

See “Discarding resource definitions” on page 27 for general information about discards.

Options

TDQUEUE(*data-value*)

specifies the 4-character name of the transient data queue that is to be removed.

Conditions

INVREQ

RESP2 values:

- 11 The queue name begins with the letter C.
- 18 The queue is not closed.
- 30 The queue is in “disable pending” status (that is, the disabling process is not completed).
- 31 The queue is not disabled.
- 200 The command was executed in a program defined with an

EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

QIDERR

RESP2 values:

- 1 The queue cannot be found.

DISCARD TERMINAL

Remove a TERMINAL definition.

DISCARD TERMINAL

►►—DISCARD TERMINAL(*data-value*)—◄◄

Conditions: INVREQ, NOTAUTH, TERMIDERR

Description

The DISCARD TERMINAL command removes the definition of a terminal from the local CICS system, so that the system no longer has access to the terminal; that is, it deletes a TERMINAL resource definition of the same name which was installed explicitly, installed automatically, or shipped by another CICS which routed a transaction to the local CICS.

To be eligible for discard, a terminal defined as local must be either a VTAM terminal or a console, it must be in out-of-service status, and it cannot be the CICS-defined error console CERR. A remote terminal cannot be in use by the local system (that is, it cannot be the principal facility of a task there). Sessions on a connection cannot be discarded with a DISCARD TERMINAL command, even if they were installed via a TERMINAL resource definition. You must use DISCARD CONNECTION instead.

CICS completes successful DISCARD TERMINAL processing with an implicit syncpoint on behalf of the issuing task, committing changes to recoverable resources made up to that point in the task. If the discard processing fails, CICS raises the INVREQ exception condition with a RESP2 value of 43, and does a SYNCPOINT ROLLBACK instead, rolling back changes to recoverable resources. In all other exception situations, however, discard processing is not attempted and neither SYNCPOINT nor SYNCPOINT ROLLBACK is issued.

See “Discarding resource definitions” on page 27 for general information about discards.

Options

TERMINAL(*data-value*)

specifies the 4-character name of the terminal whose definition is to be discarded.

Conditions

INVREQ

RESP2 values:

- 33 The terminal is an APPC session or device.
- 38 The terminal type is neither VTAM nor console.
- 39 The terminal is local and not out-of-service.
- 40 The terminal is the system error console.
- 41 The terminal is an MRO session.
- 43 Delete processing failed.
- 44 The terminal is remote and is in use locally.
- 45 The TERMINAL definition is in use.
- 46 Discard of this TERMINAL definition is already in progress.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

TERMIDERR

RESP2 values:

- 23 The terminal cannot be found

DISCARD TRANCLASS

Remove a transaction class definition.

DISCARD TRANCLASS

►►—DISCARD TRANCLASS(*data-value*)—◄◄

Conditions: INVREQ, NOTAUTH, TCIDERR

Description

The DISCARD TRANCLASS command removes the definition of a transaction class from the local CICS system. A transaction class cannot be removed while any TRANSACTION definitions belong to it.

See “Discarding resource definitions” on page 27 for general information about discards.

Options

TRANCLASS(*data-value*)

specifies the 8-character name of the transaction class that is to be removed.

In earlier releases of CICS, transaction classes were numbered from 1 through 10 rather than named, as they are now, and class definitions were implicit rather than explicit. For compatibility, CICS supplies definitions for the numbered classes, named 'DFHTCL*nn*', where *nn* is the 2-digit class number. You can discard a numbered class by using the associated name for the TRANCLASS value (DFHTCL01 for class 1, for example).

Conditions

INVREQ

RESP2 values:

- 2 The TRANCLASS definition is in use.
- 12 The transaction class cannot be discarded because installed transactions belong to it.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

TCIDERR

RESP2 values:

- 1 The transaction class cannot be found.

DISCARD TRANSACTION

Remove a transaction definition.

DISCARD TRANSACTION

►►—DISCARD TRANSACTION(*data-value*)—◄◄

Conditions: INVREQ, NOTAUTH, TRANSIDERR

Description

The DISCARD TRANSACTION command removes the definition of a transaction from the local CICS system. That is, it revokes the earlier installation of a TRANSACTION resource definition of the same name.

You cannot delete transactions supplied by CICS (names beginning with the letter C), transactions defined by the CICS system initialization table (paging transactions, for example), or transactions that are scheduled to execute at a future time or when required resources are available. Transactions already in flight are not affected; they continue to execute under the definition in force at the time they were attached.

See “Discarding resource definitions” on page 27 for general information about discards.

Options

TRANSACTION(*data-value*)

specifies the 4-character name of the transaction that is to be removed.

Conditions

INVREQ

RESP2 values:

- 4 The transaction cannot be discarded because its name begins with C.
- 13 The transaction is defined in the SIT.
- 14 The transaction is scheduled to run at a future time (in use by an interval control element).
- 15 The transaction is scheduled to run when required resources are available (in use by an automatic initiate descriptor).

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

TRANSIDERR

RESP2 values:

- 1 The transaction cannot be found.

DISCARD TSMODEL

Remove a temporary storage model definition.

DISCARD TSMODEL

►►—DISCARD TSMODEL(*data-value*)—————◄◄

Conditions: INVREQ, NOTAUTH, NOTFND

Description

The DISCARD TSMODEL command removes the definition of a temporary storage model from the local CICS system, so that the system no longer has access to the temporary storage model; that is, it revokes the earlier installation of a TSMODEL resource definition of the same name.

You can discard a TSMODEL, except those beginning with DFH, at any time. In-flight UOWs which are using such TSMODELS will complete normally.

See “Discarding resource definitions” on page 27 for general information about discards.

Options

TSMODEL (*data-value*)

specifies the 8-character name of the temporary storage model that is to be removed. .

Conditions

INVREQ

RESP2 values:

- 2 The TSMODEL definition is currently in use.
- 3 The temporary storage model cannot be discarded because its name begins with DFH.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to discard a TSMODEL definition with this name.

NOTFND

RESP2 values:

- 1 The TSMODEL does not exist.

DISCARD URIMAP

Remove a URIMAP definition from the system.

DISCARD URIMAP

►►—DISCARD URIMAP(*data-value*)—————►►

Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

The DISCARD URIMAP command removes a URIMAP definition from the system. For a URIMAP definition to be discarded, it must be disabled individually (using the SET URIMAP command). Disabling a virtual host (using the SET HOST command) does not allow the URIMAP definitions that make up the virtual host to be discarded.

See Discarding resource definitions for general information about discards.

Options

URIMAP (*data-value*)

specifies the 8-character name of the URIMAP definition that is to be removed.

Conditions

INVREQ

RESP2 values are:

- 4 The URIMAP is not disabled.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

NOTAUTH

RESP2 values are:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values are:

- 3 The URIMAP cannot be found.

DISCARD WEBSERVICE

Remove a WEBSERVICE definition.

DISCARD WEBSERVICE

►►—DISCARD WEBSERVICE(*data-value*)—◀◀

Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the **DISCARD WEBSERVICE** command to remove a WEBSERVICE resource from your CICS region.

See “Discarding resource definitions” on page 27 for general information about discards.

Options

WEBSERVICE(*data-value*)

specifies the 8-character name of the WEBSERVICE resource whose definition is to be discarded.

Conditions

INVREQ

RESP2 values:

- 6 Delete is in progress for this WEBSERVICE.
- 200 The command was executed in a program defined with an EXECUTIONSET value of DPLSUBSET or a program invoked from a remote system by a distributed program link without the SYNCONRETURN option.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

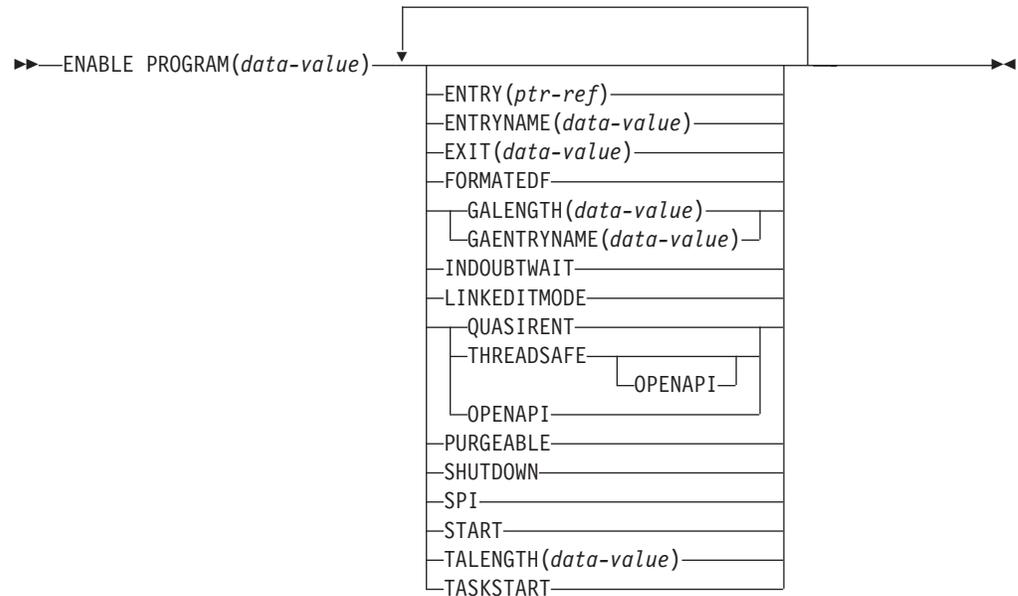
RESP2 values:

3 The WEBSERVICE cannot be found.

ENABLE PROGRAM

Enable a user exit to allow it to be invoked.

ENABLE PROGRAM



Conditions: INVEXITREQ, NOTAUTH

Description

The initial ENABLE PROGRAM command for an exit:

- Defines it as an exit to the running CICS region and names it
- Sets the initial status (whether it is available for running and the points at which it is called)
- Allocates work areas
- Loads the associated load module if required and establishes the entry point for the exit.

After the initial ENABLE command that defines the exit, you can add or remove points at which the exit is executed or change its availability dynamically with ENABLE and DISABLE commands, until you disable with the EXITALL option, which deletes the definition of the exit. See the description of the “DISABLE PROGRAM” on page 166 command for the correspondence between options on the two commands.

For programming information about exits, and a list of exit points, see the *CICS Customization Guide*. You can also read the general discussion of commands that modify exits in “Exit-related commands” on page 28.

Options

ENTRY(*ptr-ref*)

Specifies a pointer reference that contains the entry point address of the global or task-related user exit. The address you specify must be within the virtual storage range occupied by the load module named in the PROGRAM option.

The use of the ENTRY option means that the module named in the PROGRAM option has already been loaded or is permanently resident. CICS does not attempt to load the module, and also does not delete it when the user exit is disabled with EXITALL. If you omit ENTRY, CICS uses the first entry point in the load module and manages loading and deletion for you.

ENTRY is valid only on the initial ENABLE command that defines the exit.

If you specify LINKEDITMODE for a task-related user exit, the top bit of the entry address must contain the addressing mode (AMODE) indicator. The top bit is set if the exit is AMODE=31 and is zero if AMODE=24.

ENTRYNAME(*data-value*)

Specifies the 8-character name of the global or task-related user exit that is to be enabled. This name must be different from the name of any exit already established. It does not have to be defined to CICS other than by means of this command, and it need not be the name of a load module or an entry point to a load module.

If you omit ENTRYNAME, the name of the exit defaults to the name of the load module specified in the PROGRAM option.

After the initial ENABLE command that defines the exit, you must use the same combination of ENTRYNAME and PROGRAM values to identify the exit on subsequent ENABLE, DISABLE, and EXTRACT EXIT commands.

EXIT(*data-value*) (global user exits only)

Specifies the 8-character name of a global user exit point with which this exit is to be associated. When an exit is “associated” with an exit point, it is invoked when CICS reaches that particular point in its management code, provided the exit has been “started” (made available for execution). Exit points are defined and named by CICS.

You can name only one exit point on each ENABLE command. If the same exit is to be invoked from multiple exit points, you must use a separate ENABLE command for each point.

FORMATEDF (task-related user exits only)

Specifies that the exit is to be invoked at additional points (within EDF), when the exit is invoked by a task running under EDF. The additional invocations allow the exit to format EDF displays and interpret changes made by the user to fields on the EDF screen. You can turn off EDF invocations with a DISABLE command specifying FORMATEDF.

GAENTRYNAME(*data-value*)

Specifies the 8-character name of a currently enabled global or task-related user exit whose global work area is to be shared by the exit being enabled. This is the name assigned to that exit when it was defined (its ENTRYNAME if one was used or its load module name from the PROGRAM option if not).

It must own the work area (that is, GALENGTH must have been specified when it was originally enabled). CICS does not release a work area until all of the exits using it are disabled with EXITALL (no longer defined), but the owning exit must still be enabled for a new exit to share its work area.

GALENGTH and GAENTRYNAME are mutually exclusive and must be specified on the initial ENABLE command that defines the exit. If neither option is supplied, no global work area is provided.

GALENGTH (*data-value*)

specifies, as a halfword binary value, the length in bytes of the global work area that is to be provided by CICS for this exit. Valid lengths are 1 through 32767. The work area is initialized to binary zeros.

GALENGTH is valid only on the initial ENABLE command that defines the exit.

CICS does not return the address of the work area on the ENABLE command; you can use an EXTRACT EXIT command to determine it.

Note: Although the maximum GALENGTH that you can specify using this command at the terminal is 32767, there is no limit to the value you can request for GALENGTH if one of your programs issues the command. However, if more than 65535 is requested in this way, the request is truncated to the low order halfword of the requested amount. After any required truncation, if the value (which cannot now exceed 65535), exceeds 65516, an error response is issued for the INVEXITREQ condition.

INDOUBTWAIT (**task-related user exits only**)

Specifies that the task-related user exit supports the indoubt protocol.

LINKEDITMODE (**task-related user exits only**)

Specifies that the exit should be invoked in the addressing mode in which it was link-edited. If you do not specify LINKEDITMODE, it is invoked in the addressing mode of the caller. LINKEDITMODE is valid only on the initial ENABLE command that defines the exit.

You should avoid using LINKEDITMODE to force a task-related user exit to run in AMODE(24) because:

- An exit link-edited in AMODE(24) cannot be invoked from a task running with TASKDATALOC(ANY). If you attempt to do this, the task abends with CICS abend code AEZB.
- Enabling an exit for TASKSTART and LINKEDITMODE causes CICS to force all transactions to run with TASKDATALOC(BELOW) if the associated load module is link edited for AMODE(24).
- For a CICS shutdown call, CICS ignores the LINKEDITMODE attribute and invokes the exit in the addressing mode of the task that performs this shutdown function. For some types of shutdown, the addressing mode of this task is not predefined.

OPENAPI (**task-related user exits only**)

Specifies that the task-related user exit program is using non-CICS APIs. If the user application program that invokes the task-related user exit is defined as quasi-reentrant, CICS switches the user task to an L8 mode open TCB before passing control to the task-related user exit program. CICS assumes that a task-related user exit enabled with OPENAPI does not manage its own private pool of TCBs for non-CICS services, and can perform its processing on the L8 mode TCB.

If you specify OPENAPI without THREADSAFE, CICS enforces THREADSAFE by default. A task-related user exit that specifies OPENAPI must be written to threadsafe standards.

If OPENAPI is omitted, CICS assumes that the task-related user exit is either using only the CICS API, or that it performs its own TCB switch to invoke non-CICS services.

Note: You do not explicitly enable the DB2 task-related user exit, because this is done automatically for you when you start the CICS DB2 adapter. If you are connected to DB2 Version 6 or later, the exit is enabled with OPENAPI and the adapter uses L8 mode TCBs provided by CICS. The number of open TCBs permitted for this purpose is controlled by the MAXOPENTCBS system initialization parameter (see the *CICS System Definition Guide*).

For the rules that determine which calls to a task-related user exit cause the exit to be invoked on an L8 mode TCB or the QR TCB, and for other associated information, see the *CICS Customization Guide*.

PROGRAM(*data-value*)

Specifies the 8-character name of the load module containing the entry point of the exit. CICS uses the PROGRAM resource definition of this name to load the program, if necessary, and to verify that it is enabled and resides on the same CICS system as the exit. If no such definition exists, CICS attempts to build one dynamically if the system is defined to allow autoinstall of programs.

If you omit the ENTRYNAME option, CICS assumes that the name of the exit is the same as that of the load module.

PURGEABLE (task-related user exits only)

Allows tasks that have entered a CICS wait state and are active in the task-related user exit, to be purged. The task-related user exit must be written to correctly process the purged response from the wait if this option is to be used. You can turn this off with a DISABLE command specifying PURGEABLE.

QUASIRENT

Specifies that the global user exit program or task-related user exit program is quasi-reentrant, and relies on the serialization provided by CICS when accessing shared resources. The user exit program is restricted to the CICS permitted programming interfaces, and must comply with CICS quasi-reentrancy rules. CICS always invokes a quasi-reentrant user exit under the QR TCB.

A task-related user exit program is allowed to use MVS services. If it does so, it must switch to its own private TCB before issuing calls to these services, and switch back again before returning to its caller.

SHUTDOWN (task-related user exits only)

Specifies that the exit is to be invoked during CICS shutdown processing. You can turn off the invocation with a DISABLE command specifying SHUTDOWN.

SPI (task-related user exits only)

Specifies that the task-related user exit is to be invoked if an INQUIRE EXITPROGRAM command which names it specifies the CONNECTST or QUALIFIER option, or both.

The task-related user exit program is invoked with an SPI call, allowing it to return CONNECTST and QUALIFIER information to the inquiring program. For details of RMI SPI calls, see the *CICS Customization Guide*.

START

Indicates that the exit is available for execution. You can turn availability on

and off with ENABLE commands (specifying START) and DISABLE commands (specifying STOP), but the exit starts out in stopped mode and is not available until the first ENABLE with START.

When a STOPped task-related user exit gets invoked, the invoking code gets an AEY9 abend code. There is no corresponding error for global user exits, however, because CICS invokes only those exits associated with an exit point which are also available for execution (not stopped).

When a single global user exit is to be associated with several exit points, the START option allows you to delay execution of the exit until all the required ENABLE commands have been issued. You can, however, associate more exit points with the exit *after* it has been started.

TALENGTH(*data-value*) (task-related user exits only)

Specifies, as a halfword binary value, the length in bytes of the work area that CICS provides for each task that uses the exit. Valid lengths are 1 through 32767. CICS allocates the work area and initializes it to binary zeros before the first use of the exit by the task, and releases it at task end. If you do not specify TALENGTH, CICS does not create task work areas.

TASKSTART (task-related user exits only)

Specifies that the exit is to be invoked at the start of every task. The exit is also invoked at end of task, but you can turn off this invocation within the exit if you want. (The task that logs off an autoinstalled terminal in an MRO environment is an exception; it does not invoke the exit.)

The TASKSTART option is independent of the START option, but you should turn on START before or at the same time as TASKSTART, to avoid invoking the exit when it is not available for execution. In addition, you must not code the TASKSTART option on any ENABLE command that can be executed before the recovery part of CICS initialization.

You can turn off these invocations with a DISABLE command specifying TASKSTART.

THREADSAFE

Specifies that the global user exit program or task-related user exit program is written to threadsafe standards, and takes into account the possibility that, when accessing shared resources, other programs may be executing concurrently and attempting to modify the same resources. It uses appropriate serialization techniques when accessing any shared resources.

A threadsafe user exit program must be able to run under whichever TCB CICS invokes it. This could be either the QR TCB or an open TCB. (For task-related user exits only, if OPENAPI is also specified CICS will always invoke the task-related user exit under an L8 open TCB.)

Conditions

INVEXITREQ

The INVEXITREQ condition of the ENABLE command is indicated by X'80' in the first byte of EIBRCODE. The exact cause of the error can be determined by examining the second and third bytes of EIBRCODE.

X'808000'

The load module named in the PROGRAM option has not been defined to CICS and could not be autoinstalled, or is not in the load library, or has been disabled, or is defined as remote, or does not contain the address specified in the ENTRY option.

- X'804000'**
The name specified in the EXIT option is not a valid global user exit point.
- X'802000'**
The exit is already enabled. ENTRY, LINKEDITMODE, TALENGTH, GAENTRY, GALENGTH, QUASIRENT, and THREADSAFE are valid only on the initial ENABLE command that defines the exit.
- X'801000'**
The exit is already associated with the exit point specified in the EXIT option.
- X'800800'**
The exit specified in the GAENTRYNAME option is not enabled.
- X'800400'**
The exit specified in the GAENTRYNAME option does not own a work area.
- X'800040'**
The length specified in GALENGTH option exceeds the maximum allowed of 65516.

NOTAUTH

RESP2 values:

- 100** The user associated with the issuing task is not authorized to use this command.
- 101** The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

Enabling global user exits

These examples show how you can use the **ENABLE PROGRAM** command to enable global user exits.

Example 1

```
EXEC CICS ENABLE PROGRAM('EP1') ENTRYNAME('EP1')
      EXIT('XFCREQ') START
```

Example 1 defines exit EP1, tells CICS that EP1 is to be invoked from exit point XFCREQ, and makes EP1 available for execution. No global work area is obtained. CICS loads the EP module if necessary.

Example 2

```
EXEC CICS ENABLE PROGRAM('EP2') EXIT('XMNOUT')
      START ENTRY(EADDR) GALENGTH(500)
```

Example 2 defines an exit named EP2 (named by default from its load module). This module is already loaded, and the entry point for the exit is in EADDR. The exit is to be executed at exit point XMNOUT, and it is available for execution. A global work area of 500 bytes, which is to be owned by EP2, is obtained.

Example 3

```
EXEC CICS ENABLE PROGRAM('EP3') EXIT('XTDOUT')
      GAENTRYNAME('EP2')
EXEC CICS ENABLE PROGRAM('EP3') EXIT('XTDIN')
EXEC CICS ENABLE PROGRAM('EP3') EXIT('XTDREQ') START
```

The first command of Example 3 defines exit EP3; it is associated with exit point XTDOUT. CICS loads module EP3 if necessary. EP3 is to use the work area that is owned by exit EP2. (This assumes that the ENABLE command in Example 2 has already been issued.)

The second command says that EP3 is also associated with exit point XTDIN. The third command says that EP3 is associated with exit point XTDREQ, and makes the exit available for execution. EP3 is now invoked from all of these exit points, and it can use EP2's work area on any of those invocations.

Enabling task-related user exits

The following example shows you how to use the **ENABLE PROGRAM** command to enable task-related user exits.

Example

```
EXEC CICS ENABLE PROGRAM('EP9')
      TALENGTH(750) ENTRYNAME('RM1') GALENGTH(200)

EXEC CICS ENABLE PROGRAM('EP9')
      ENTRYNAME('RM1') START
```

The first command defines the task-related user exit RM1, loads EP9 (the load module executed initially) if it is not already resident, and allocates a 200-byte global work area to the exit. It also schedules the allocation of a further 750-byte work area for each task that invokes RM1. The second command makes the exit available for execution.

EXTRACT EXIT

Obtain the address and length of a global work area.

EXTRACT EXIT

```
▶▶—EXTRACT EXIT PROGRAM(data-value)—▶
      └─ENTRYNAME(data-value)—┘
▶—GALENGTH(data-area)—GASET(ptr-ref)—▶▶
```

Conditions: INVEXITREQ, NOTAUTH

Description

The EXTRACT EXIT command obtains the address and length of the global work area that is owned by, or shared by, a user exit.

Note: To enable the use of application programs written for earlier releases that specify DSNCEXT1 or DSN2EXT1 on the EXTRACT EXIT command to inquire on the status of the CICS-DB2 interface, CICS automatically substitutes the correct name, DFHD2EX1. CICS does this by setting argument 1 in the parameter list to address the new name, and no application program storage is altered. This allows existing application programs to work unchanged.

Options

ENTRYNAME(*data-value*)

specifies the 8-character name of the global or task-related user exit for which

you want global work area information. If you omit ENTRYNAME, CICS assumes that the name of the exit is the same as the name of the load module given in the PROGRAM option. Therefore, you must use the same combination of ENTRYNAME and PROGRAM values as was specified on the ENABLE command that defined the exit.

GALENGTH(*data-area*)

returns the length in bytes of the global work area, in halfword binary form.

Note: If a GALENGTH greater than 32767 has been defined (see GALENGTH for ENABLE PROGRAM for details), the response to this command reflects that higher value as follows:

- If you issued the EXTRACT EXIT command at your terminal, the response shows a negative value for GALENGTH.
- If you issued the EXTRACT EXIT command from a program, the high order bit of the response for GALENGTH is set. You must allow for this possibility when deciding what operation to next perform on the returned value.

GASET(*ptr-ref*)

returns the address of the global work area.

PROGRAM(*data-value*)

specifies the name of the load module containing the entry point of the exit. This name is also used as the name of the exit when ENTRYNAME is not specified; see the ENTRYNAME option.

Conditions

INVEXITREQ

The INVEXITREQ condition of the EXTRACT EXIT command is indicated by X'80' in the first byte of EIBRCODE. The exact cause of the error can be determined by examining the second and third bytes of EIBRCODE. For further information on EIBRCODE, see Appendix B, "EXEC interface block (EIB) response and function codes," on page 825.

X'800200'

The exit is not enabled.

X'800400'

The exit has no global work area.

X'808000'

The load module named in the PROGRAM option is not the same as the one used when the exit specified in the ENTRYNAME option was enabled.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

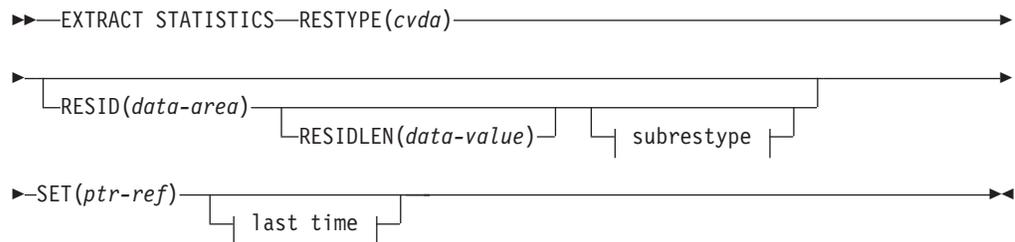
101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

EXTRACT STATISTICS

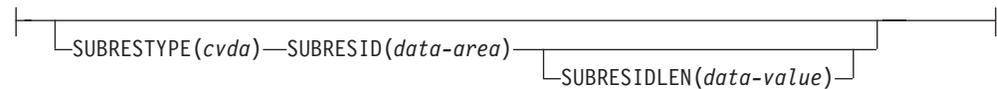
Retrieve the current statistics for a single resource, or global statistics for a class of resources. EXTRACT STATISTICS deals only with the ATOMSERVICE, BUNDLE, CAPTURESPEC, DOCTEMPLATE, EPADAPTER, EVENTBINDING, EVENTPROCESS, IPCONN, JVMSERVER, LIBRARY, MQCONN, PIPELINE, PROGRAMDEF, URIMAP, WEBSERVICE, and XMLTRANSFORM resources.

To collect statistics for other resources, use COLLECT STATISTICS in CICS System Programming Reference. EXTRACT STATISTICS performs a function equivalent to COLLECT STATISTICS for the resources named above.

Extract STATISTICS



subrestype:



last time:



hours:



minutes:



seconds:



Conditions: INVREQ, IOERR, LENGERR, NOTAUTH, NOTFND

Description

The **EXTRACT STATISTICS** command returns to the invoking application the current statistics for a particular resource, or global statistics for the resources of a given type.

The statistics that CICS returns are those that have been accumulated after the expiry of the last statistics extraction interval, end-of-day expiry, or requested reset. Statistics already written to the SMF data set cannot be accessed. The **EXTRACT STATISTICS** command does not cause the statistics counters to be reset.

CICS obtains enough storage for the data returned from this command and returns a pointer to this area. The first two bytes of the area contain its length. This storage can be reused by subsequent **EXTRACT STATISTICS** commands, so you must store elsewhere any data that is required beyond the next issue of the command. CICS releases this storage at task termination.

Table 2 contains Product-sensitive Programming Interface information.

Not all resource types provide both global and specific statistics. Table 2 tells you which statistics are available for each resource type and gives the copybook name for each set of available statistics. The copybooks define the format of the returned statistics. Where no copybook name is given in the global statistics column, global statistics are unavailable for the resource type; similarly, where the specific statistics column contains no entry, you cannot get statistics for an individual resource.

Table 2. Resource types and statistics

Resource type	CVDA	RESIDLEN	Statistic type	Subresource type	CVDA	SUBRESIDLEN	Global statistics	Specific statistics
ATOMSERVICE	1179	8	ATOMSERVICE	—	—	—	—	DFHW2RDS
BUNDLE	1180	8	BUNDLE	—	—	—	—	DFHRLRDS
DOCTEMPLATE	1145	8	DOCTEMPLATE	—	—	—	—	DFHDHDDS
EPADAPTER	1196	32	EPADAPTER	—	—	—	—	DFHEPRDS
EVENTBINDING	1191	32	EVENTBINDING	—	—	—	DFHECGDS	DFHECRDS
EVENTBINDING	1191	32	EVENTBINDING	CAPTURESPEC	1195	32	—	DFHECCDS
EVENTPROCESS	1192	—	EVENTPROCESS	—	—	—	DFHEPGDS	—
IPCONN	1176	8	IPCONN	—	—	—	—	DFHISRDS
JVMSEVER	1193	8	JVMSEVER	—	—	—	—	DFHSJSDS
LIBRARY	1177	8	LIBRARY	—	—	—	—	DFHLDBDS
MQCONN	1175	—	MQCONN	—	—	—	DFHMQGDS	—
PIPELINE	1124	8	PIPELINE	—	—	—	—	DFHPIRDS
PROGRAMDEF	1178	8	PROGRAMDEF	—	—	—	—	DFHPGDDS
URIMAP	1173	8	URIMAP	—	—	—	DFHWBGDS	DFHWBRDS
WEBSERVICE	1174	32	WEBSERVICE	—	—	—	—	DFHPIWDS
XMLTRANSFORM	1194	32	XMLTRANSFORM	—	—	—	—	DFHMLRDS

Copybooks are provided in ASSEMBLER, C, COBOL, and PL/I.

The names of the copybooks are the same in each language. You can find them in the following libraries:

ASSEMBLER	CICSTS42.CICS.SDFHMAC
C	CICSTS42.CICS.SDFHC370
COBOL	CICSTS42.CICS.SDFHCOB

Note: Some of the copybooks contain packed fields. Before these fields are used, check them for hexadecimal zeros. The COBOL versions of the fields have been redefined as numeric with a suffix of -R for this purpose.

For further information about these copybooks, see in the *CICS Performance Guide*.

Options

LASTRESET (*data-area*)

Returns a 4-byte packed decimal field giving the time at which the counters for the requested statistics were last reset. This time is usually when the last interval expired. The last reset time is always returned in local time.

The reset time has two formats:

- A composite format (packed decimal format 0hhmss+), which you obtain by using the LASTRESET option.
- Separate hours, minutes, and seconds, which you obtain by specifying the LASTRESETHRS, LASTRESETMIN, and LASTRESETSEC options respectively.

LASTRESETHRS (*data-area*)

Returns a fullword binary field giving the hours component of the time at which the counters for the requested statistics were last reset; see the LASTRESET option.

LASTRESETMIN (*data-area*)

Returns a fullword binary field giving the minutes component of the time at which the counters for the requested statistics were last reset; see the LASTRESET option.

LASTRESETSEC (*data-area*)

Returns a fullword binary field giving the seconds component of the time at which the counters for the requested statistics were last reset; see the LASTRESET option.

RESTYPE (*cvda*)

Requests statistics for a particular resource type depending on the CVDA value supplied. Valid CVDA values are as follows:

ATOMSERVICE

Request statistics for an ATOMSERVICE resource; RESID identifies the particular ATOMSERVICE resource definition.

BUNDLE

Request statistics for a BUNDLE resource; RESID identifies the particular BUNDLE resource definition.

DOCTEMPLATE

Request statistics for a document template; RESID identifies the particular DOCTEMPLATE resource definition.

EPADAPTER

Request statistics for an EPADAPTER resource; RESID identifies the particular EPADAPTER resource definition.

EVENTBINDING

Request statistics for a particular EVENTBINDING resource; RESID identifies the particular EVENTBINDING resource definition.

|
|
|

EVENTPROCESS

Request global statistics on the event processing domain.

IPCONN

Request statistics for an IPCONN resource; RESID identifies the particular IPCONN resource definition.

JVMSEVER

Request statistics for a JVMSEVER resource; RESID identifies the particular JVMSEVER resource definition.

LIBRARY

Request statistics for a LIBRARY resource; RESID identifies the particular LIBRARY resource definition.

MQCONN

Request statistics for a WebSphere MQ connection.

PIPELINE

Request statistics for a PIPELINE resource; RESID identifies the particular PIPELINE resource definition.

URIMAP

Request statistics for a URIMAP resource; RESID identifies the particular URIMAP resource definition.

WEBSERVICE

Request statistics for a WEBSERVICE resource; RESID identifies the particular WEBSERVICE resource definition.

XMLTRANSFORM

Request statistics for an XMLTRANSFORM resource; RESID identifies the particular XMLTRANSFORM resource definition.

RESID(*data-area*)

Specifies the name of the resource for which statistics are being extracted. The absence of this keyword means that global statistics are to be extracted. RESID is a character field.

RESIDLEN(*data-value*)

Specifies the length of the RESID data area. If omitted, the default value is the length given in Table 2 on page 208.

SET(*ptr-ref*)

Specifies a pointer reference to be set to the address of the data area containing the returned statistics. The first 2 bytes of the data area contain the length of the data area in halfword binary form.

SUBRESTYPE(*cvda*)

Requests statistics for a particular resource type depending on the CVDA value supplied. The **subrestype** parameter is optional; for usage see Table 2 on page 208. Valid CVDA values are as follows:

CAPTURESPEC

Request statistics for a capture specification.

SUBRESID(*data-area*)

Specifies the name of the resource for which statistics are being extracted. The absence of this keyword means that statistics for the specified RESTYPE are to be extracted. SUBRESID is a character field.

SUBRESIDLEN(*data-value*)

Specifies the length of the SUBRESID data area. If omitted, the default value is the length given in Table 2 on page 208.

Conditions**INVREQ**

RESP2 values:

- 5 An invalid RESTYPE has been specified. Valid types are listed in Table 2 on page 208.
- 6 A mandatory RESID has not been specified for the requested RESTYPE.
- 8 An invalid SUBRESTYPE has been specified. Valid types are listed in Table 2 on page 208.
- 9 A mandatory SUBRESID has not been specified for the requested SUBRESTYPE.
- 11 An invalid RESTYPE and SUBRESTYPE combination has been specified. Valid types are listed in Table 2 on page 208.

IOERR

RESP2 values:

- 3 The requested statistics area was not functioning. This problem occurs if, for example, statistics control blocks are overwritten.

LENGERR

RESP2 values:

- 7 An invalid RESIDLEN was supplied for the requested RESID.
- 10 An invalid SUBRESIDLEN was supplied for the requested SUBRESID.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

NOTFND

RESP2 values:

- 1 The requested resource cannot be found; for example, if the WEBSERVICE you specify is not known to CICS.
- 2 The type of resource is not defined in the CICS system; for example, if FEPI statistics are requested with POOL or NODE when the **FEPI** system initialization parameter specifies NO.

Examples

```
EXEC CICS EXTRACT STATISTICS URIMAP
or
EXEC CICS EXTRACT STATISTICS RESTYPE(1173)
or
EXEC CICS EXTRACT STATISTICS RESTYPE(DFHVALUE(URIMAP))
```

CICS provides a sample EXTRACT STATISTICS application, DFH0STAT, that uses the options described in this topic. This set of programs illustrates ways of using

the EXTRACT STATISTICS and INQUIRE commands to produce information about a CICS system. The reports include a CICS and MVS storage analysis that can be used as an aid to specifying the DSA LIMIT parameters.

See the *CICS Performance Guide* for information about installing and operating the DFH0STAT application. The source code for the application is in CICSTS42.CICS.SDFHSAMP.

INQUIRE ASSOCIATION

Retrieve association information for a specified task from its association data control block (ADCB).

INQUIRE ASSOCIATION

►►—INQUIRE ASSOCIATION(*data-value*)—►►
└─ options ─┘

Conditions: INVREQ, NOTAUTH, TASKIDERR

This command is threadsafe.

Options:

APPLDATA(<i>data-area</i>)
APPLID(<i>data-area</i>)
CLIENTIPADDR(<i>data-area</i>)
CLIENTLOC(<i>data-area</i>)
CLIENTPORT(<i>data-area</i>)
CLNTIPFAMILY(<i>cvda</i>)
DNAME(<i>data-area</i>)
FACILNAME(<i>data-area</i>)
FACILTYPE(<i>cvda</i>)
INITUSERID(<i>data-area</i>)
IPCONN(<i>data-area</i>)
IPFAMILY(<i>cvda</i>)
LUNAME(<i>data-area</i>)
MVSIMAGE(<i>data-area</i>)
NETID(<i>data-area</i>)
ODADPTRID(<i>data-area</i>)
ODADPTRDATA1(<i>data-area</i>)
ODADPTRDATA2(<i>data-area</i>)
ODADPTRDATA3(<i>data-area</i>)
ODAPPLID(<i>data-area</i>)
ODCLNTIPADDR(<i>data-area</i>)
ODCLNTPORT(<i>data-area</i>)
ODFACILNAME(<i>data-area</i>)
ODFACILTYPE(<i>cvda</i>)
ODIPFAMILY(<i>cvda</i>)
ODLUNAME(<i>data-area</i>)
ODNETID(<i>data-area</i>)
ODNETWORKID(<i>data-area</i>)
ODSTARTTIME(<i>data-area</i>)
ODTASKID(<i>data-area</i>)
ODTRANSID(<i>data-area</i>)
ODUSERID(<i>data-area</i>)
PHAPPLID(<i>data-area</i>)
PHCOUNT(<i>data-area</i>)
PHNETWORKID(<i>data-area</i>)
PHSTARTTIME(<i>data-area</i>)
PHTASKID(<i>data-area</i>)
PHTRANSID(<i>data-area</i>)
PROGRAM(<i>data-area</i>)
REALM(<i>data-area</i>)
SERVERIPADDR(<i>data-area</i>)
SERVERPORT(<i>data-area</i>)
SRVRIPFAMILY(<i>cvda</i>)
STARTTIME(<i>data-area</i>)
TCPIPJOB(<i>data-area</i>)
TCPIPSERVICE(<i>data-area</i>)
TCPIPZONE(<i>data-area</i>)
TRNGRPID(<i>data-area</i>)
TRANSACTION(<i>data-area</i>)
USERCORRDATA(<i>data-area</i>)
USERID(<i>data-area</i>)

For more information about the use of CVDAs, see CICS-value data areas (CVDAs) in CICS System Programming Reference.

Description

The INQUIRE ASSOCIATION command retrieves information about the way a task was started, based on a task number.

Association records are identified by task numbers. Therefore, the input data, specified on the ASSOCIATION option of the INQUIRE command, is the task number. The association data is retrieved from the association data control block (ADCB) of the specified task.

The association data control block is built during task attach processing. It might contain information about another CICS task that acted as the point of origin for this task.

Use the INQUIRE ASSOCIATION command to inquire about the association data of a single task in the local region. Browsing is not supported.

Options

APPLDATA(*data-area*)

Returns the 40-character value of the application data associated by CICS with the socket that received the request that started this task. If the task was not started through a socket, APPLDATA is blank.

The 40-character application data consists of these bytes:

A 24-byte prefix owned by the Sockets domain

Bytes 01-03

“DFH”

Byte 04

I Inbound (listen and accept)
O Outbound (connect)

Bytes 05-12

The APPLID of this region

Bytes 13-16

The ID of the transaction that created the socket:

CIEP ECI inbound
CIRR IIOP inbound
CISC IPIC outbound
CISS IPIC inbound
CWYN HTTP inbound
CWYU USER inbound
xxx HTTP outbound
xxx IIOP outbound

Bytes 17-24

The network protocol: one of ECI, HTTP, IIOP, IPIC, or USER

A 16-byte suffix owned by the using domain

The contents of the suffix depends on the state of the connection:

The TCPIPSERVICE is listening on the socket

Bytes 25-32

The TCPIPSERVICE name

Bytes 33-40

The first 8 bytes of the TCPIPSERVICE description

After the IPCONN has been acquired**Bytes 25-32**

The IPCONN name

Bytes 33-40

The APPLID of the partner region

Default for outbound connections**Bytes 25-40**

Blank

This data can be used to correlate CICS connection information with z/OS Communication Server connection information.

APPLID(*data-area*)

Returns the 8-character APPLID of the CICS region in which this task is running.

ASSOCIATION(*data-value*)

Specifies the 4-byte number of the task for which you want to retrieve association data.

CLIENTIPADDR(*data-area*)

Returns, into a 39-character area, the IP address of the TCP/IP client that requested this task to start. When the CLNTIPFAMILY option returns IPv4, the returned address is a 15-character dotted decimal IPv4 address, padded with blanks. When CLNTIPFAMILY returns IPv6, the address returned is a 3- to 39-character colon hexadecimal IPv6 address, padded with blanks. If this task was not started from a TCP/IP client, CLIENTIPADDR returns 0.0.0.0 and CLNTIPFAMILY returns NOTAPPLIC. If the source of this task has not yet been determined, CLIENTIPADDR returns 0.0.0.0 and CLNTIPFAMILY returns UNKNOWN.

You can specify IPv4 and IPv6 addresses in a number of acceptable formats. See the *CICS Internet Guide* for more information about address formats.

CLIENTLOC(*data-area*)

Returns a 32-character area that represents the SO_CLUSTERCONNTYPE socket option returned by z/OS Communications Server for the facility in the FACILNAME option. The binary format of SO_CLUSTERCONNTYPE is converted to characters in CLIENTLOC and displayed as either zeros or ones. The CLIENTLOC option represents the current socket, unless the value in the FACILTYPE option is IPIC, in which case CLIENTLOC is taken from the CLIENTLOC value for the IPCONN. For details, see INQUIRE IPCONN. See the z/OS 1.11 Communications Server IP Sockets Application Programming Interface Guide, http://publib.boulder.ibm.com/infocenter/zos/v1r9/index.jsp?searchWord=SO_CLUSTERCONNTYPE&tab=search, in the z/OS 1.11 information center for a description of SO_CLUSTERCONNTYPE and an explanation of the bit settings.

CLIENTPORT(*data-area*)

Returns, in fullword binary form, the number of the port that the TCP/IP stack used to send the request that resulted in this task being attached. If the task was not started in this way, CLIENTPORT returns zero.

CLNTIPFAMILY(*cvda*)

Returns a value indicating the form of TCP/IP addressing used by this task. The CVDA values are as follows:

IPV4 The address is specified in IPv4 dotted decimal address format.

IPV6 The address is specified in IPv6 colon hexadecimal address format.

NOTAPPLIC

0.0.0.0 is specified in the CLIENTIPADDR option and the task was not started from a TCP/IP client.

UNKNOWN

0.0.0.0 is specified in the CLIENTIPADDR option and the source of this task has not yet been determined.

DNAME(*data-area*)

Returns the 1- to 246-character distinguished name padded with trailing ASCII blanks. Distinguished names are represented in UTF-8 encoding. If a distinguished name is not available for the task, DNAME returns ASCII blanks.

FACILNAME(*data-area*)

Returns the 8-character name of the facility associated with the initiation of this task. If the task was started by an unnamed facility, FACILNAME returns blanks.

FACILTYPE(*cvda*)

Returns a CVDA value identifying the type of facility that initiated this task. CVDA values are as follows:

APPC LU 6.2 (APPC) connection

BRIDGE

3270 bridge facility virtual terminal

EVENT

Event processing channel

IIOP Internet Inter-ORB Protocol (IIOP) request receiver object

IPECI IP ECI Client Conversation session

IPIC IP interconnectivity session (IPCONN)

LU61 LU 6.1 session

MRO MRO session

NONE

No facility is associated with this task

RRSUR

Recovery Manager Unit of Recovery

RZINSTOR

Request stream (RZ) instore transport client

SCHEDULER

Scheduler timer request entry

SOCKET

Socket domain session entry

START

Non terminal-related START element

STARTTERM

Terminal-related START element

TERMINAL

Terminal entry

TRANDATA

Transient data destination entry

UNKNOWN

The facility type is unknown

WEB CICS web support session
XMRUNTRAN
CICS business transaction services (BTS) activity

INITUSERID(*data-area*)

Returns the 8-character user ID of the initiating task; that is, the task that caused this one to be attached.

IPCONN(*data-area*)

Returns the 8-character name of any IPIC connection that was used to receive a request that resulted in this task starting. If the task was not started in this way, IPCONN returns blanks. This field contains a nonblank value only when the FACILTYPE is IPIC.

IPFAMILY(*cvda*)

Replaced by the SRVRIPFAMILY option, which supports IPv6 addressing. IPFAMILY is maintained for existing programs only. Returns a CVDA value indicating the form of TCP/IP addressing used by this task. CVDA values are as follows:

IPV4 The request that caused CICS to initiate this task arrived at a TCPIP SERVICE resource that used an IPv4 address.

IPV6 The request that caused CICS to initiate this task arrived at a TCPIP SERVICE resource that used an IPv6 address.

NOTAPPLIC

No TCP/IP client is associated with this task.

LUNAME(*data-area*)

Returns the 8-character network name of the terminal from which this task was started. If the task was started from an IPIC (IPCONN), ISC over SNA (APPC), or MRO session, LUNAME returns the APPLID of the remote region. If the task was not started from a terminal, nor from an IPCONN, APPC, or MRO session, LUNAME returns blanks. For OTS transactions, LUNAME returns blanks.

MVSIMAGE(*data-area*)

Returns the 8-character name of the MVS image associated with the TCPIP SERVICE used to receive a request that resulted in this task starting. If the task was not started in this way, MVSIMAGE returns blanks.

This function depends on Communication Server TCP/IP Network Access Control support being activated and the CLIENTIPADDRESS being configured into a Network Security Zone.

NETID(*data-area*)

Returns the 8-character network ID of the terminal from which this task was started.

ODADPTRID(*data-area*)

Returns, in a 64-character area, the data that was added to the origin data by the adapter. This field is created when the originating task is started. If the task was not started by using an adapter, or if it was and the adapter did not set this value, ODADPTRID returns blanks.

ODADPTRDATA1(*data-area*)

Returns, in a 64-character area, the data that was added to the origin data by the adapter. This field is created when the originating task is started. If the task was not started by using an adapter, or if it was and the adapter did not set

this value, ODADPTRDATA1 returns blanks. ODADPTRDATA1 also returns blanks if the adapter set a value for this field, but did not set an adapter identifier.

ODADPTRDATA2 (*data-area*)

Returns, in a 64-character area, the data that was added to the origin data by the adapter. This field is created when the originating task is started. If the task was not started by using an adapter, or if it was and the adapter did not set this value, ODADPTRDATA2 returns blanks. ODADPTRDATA2 also returns blanks if the adapter set a value for this field, but did not set an adapter identifier.

ODADPTRDATA3 (*data-area*)

Returns, in a 64-character area, the data that was added to the origin data by the adapter. This field is created when the originating task is started. If the task was not started by using an adapter, or if it was and the adapter did not set this value, ODADPTRDATA3 returns blanks. ODADPTRDATA3 also returns blanks if the adapter set a value for this field, but did not set an adapter identifier.

ODAPPLID (*data-area*)

Returns the 8-character APPLID taken from the origin descriptor associated with this task.

ODCLNTIPADDR (*data-area*)

Returns, into a 39-character area, the IP address of the TCP/IP client that requested the originating task to start. When ODIPFAMILY returns IPV6, the address returned is a 3- to 39-character colon hexadecimal IPv6 address, padded with blanks. If the originating task was not started from a TCP/IP client, ODCLNTIPADDR returns 0.0.0.0 and ODIPFAMILY returns NOTAPPLIC. If this task was not started from a TCP/IP client, ODCLNTIPADDR returns 0.0.0.0. If the source of this task has not yet been determined, ODCLNTIPADDR returns 0.0.0.0 and ODIPFAMILY returns UNKNOWN.

ODCLNTPORT (*data-area*)

Returns, in fullword binary form, the number of the port that the TCP/IP stack used to send the request that resulted in the originating task being attached. If the originating task was not started in this way, ODCLNTPORT returns zero.

ODFACILNAME (*data-area*)

If the facility associated with the initiation of the originating task is a transient data queue, a terminal, or a system, ODFACILNAME returns the 8-character name of the facility. If the originating task was not started in any of these ways, ODFACILNAME returns blanks.

ODFACILTYPE (*cvda*)

Returns a CVDA value identifying the type of facility that initiated the originating task that is associated with this task. CVDA values are as follows:

APPC LU 6.2 (APPC) connection
BRIDGE
3270 bridge facility virtual terminal
EVENT
Event processing channel
IIOP Internet Inter-ORB Protocol (IIOP) request receiver object
IPECI IP ECI Client Conversation session
IPIC IP interconnectivity session (IPCONN)
LU61 LU 6.1 session
MRO MRO session

NONE
No facility is associated with this task

RRSUR
Recovery Manager Unit of Recovery

RZINSTOR
Request stream (RZ) instore transport client

SCHEDULER
Scheduler timer request entry

SOCKET
Socket domain session entry

START
Non terminal-related START element

STARTTERM
Terminal-related START element

TERMINAL
Terminal entry

TRANDATA
Transient data destination entry

UNKNOWN
The facility type is unknown

WEB CICS web support session

XMRUNTRAN
CICS business transaction services (BTS) activity

ODIPFAMILY (*cvda*)

Returns a value indicating the form of TCP/IP addressing used by the originating task. CVDA values are as follows:

IPV4 The address is specified in IPv4 dotted decimal address format.

IPV6 The address is specified in IPv6 colon hexadecimal address format.

NOTAPPLIC

0.0.0.0 is specified in the ODCLNTIPADDR option and the task was not started from a TCP/IP client.

UNKNOWN

0.0.0.0 is specified in the ODCLNTIPADDR option and the source of the task has not been determined.

ODLUNAME (*data-area*)

Returns the 8-character network logical unit name of the terminal from which the originating task was started. If the originating task was started from an IPIC (IPCONN), ISC over SNA (APPC), or MRO session, ODLUNAME returns the network name of the remote region. If the originating task was not started from a terminal, nor from an IPCONN, APPC, or MRO session, ODLUNAME returns blanks. For OTS transactions, ODLUNAME returns blanks.

ODNETID (*data-area*)

Returns the 8-character network ID of the terminal (terminal, APPC peer, or similar device) from which the originating task was started.

ODNETWORKID (*data-area*)

Returns the 8-character network qualifier for the origin region APPLID on which the task ran.

ODSTARTTIME (*data-area*)

Returns a 21-character representation of the time when the originating task was started. The time is in the form *yyymmddhhmmss.ssssss*.

ODTASKID(*data-area*)

Returns the 4-byte packed decimal identifier of the originating task that is associated with this task.

ODTRANSID(*data-area*)

Returns the 4-character name of the transaction under which the originating task ran.

ODUSERID(*data-area*)

Returns the 8-character user ID under which the originating task ran.

PHAPPLID(*data-area*)

Returns the 8-character APPLID from previous hop data. If the specified task was initiated by a task in another CICS region, PHAPPLID contains the APPLID of the other CICS region, or spaces if it was not initiated in this way. See Previous hop data characteristics for more information about previous hop data.

PHCOUNT(*data-area*)

Returns, in fullword binary form, the number of times there has been a request from one CICS region to another to initiate a task with which this task is associated, or zero if there have been no such requests.

PHNETWORKID(*data-area*)

Returns the 8-character network qualifier from previous hop data. If the specified task was initiated by a task in another CICS region, PHNETWORKID contains the network qualifier for the APPLID of the other CICS region or spaces if it was not initiated in this way.

PHSTARTTIME(*data-area*)

Returns a 21-character representation of the task start time from previous hop data. The time is in the form *yyymmddhhmmss.ssssss*. If the specified task was initiated by a task in another CICS region, PHSTARTTIME contains the start time of the task in the other CICS region, or spaces if it was not initiated in this way.

PHTASKID(*data-area*)

Returns the 4-byte packed decimal identifier from previous hop data. If the specified task was initiated by a task in another CICS region, PHTASKID contains the identifier of the task in the other CICS region, or packed decimal zero if it was not initiated in this way.

PHTRANSID(*data-area*)

Returns the 4-character name of a transaction from previous hop data. If the specified task was initiated by a task in another CICS region, PHTRANSID contains the transaction name of the task in the other CICS region, or spaces if it was not initiated in this way.

PROGRAM(*data-area*)

Returns the 8-character name of the first program called by a task running this transaction.

REALM(*data-area*)

Returns the 1- to 255-character realm name in UTF-8 encoding, padded with trailing ASCII blanks. The realm is a component of a distributed identity and defines the region where a security ID applies.

SERVERIPADDR(*data-area*)

Returns, into a 39-character area, the IP address of the TCP/IP service that scheduled this task. When the IPFAMILY option returns IPv4, the returned address is a 15-character dotted decimal IPv4 address, padded with blanks.

When SRVRIPFAMILY returns IPV6, it is a 3- to 39-character colon hexadecimal IPv6 address, padded with blanks. If this task was not started from a TCP/IP service, SERVERIPADDR returns 0.0.0.0 and SRVRIPFAMILY returns NOTAPPLIC. If the source of this task has not yet been determined, SERVERIPADDR returns 0.0.0.0 and SRVRIPFAMILY returns UNKNOWN.

You can specify IPv4 and IPv6 addresses in a number of acceptable formats. See the *CICS Internet Guide* for more information about address formats.

SERVERPORT (*data-area*)

Returns, in fullword binary form, the number of the port on which the TCPIP SERVICE resource that received the request that resulted in this task being attached is listening. If the task was not started in this way, SERVERPORT returns zero.

SRVRIPFAMILY (*cvda*)

Replaces the IPFAMILY option. SRVRIPFAMILY returns a value indicating the form of IP addressing used by this task. CVDA values are as follows:

IPV4 The address is specified in IPv4 dotted decimal address format.

IPV6 The address is specified in IPv6 colon hexadecimal address format.

NOTAPPLIC

0.0.0.0 is specified in the SERVERIPADDR option and the task was not started from a TCP/IP client.

UNKNOWN

0.0.0.0 is specified in the SERVERIPADDR option and the source of the task has not been determined.

STARTTIME (*data-area*)

Returns a 21-character representation of the time when this task was started. The time is in the form *yyymmddhhmmss.ssssss*.

TCPIPJOB (*data-area*)

Returns the 8-character name of the TCP/IP job associated with the IPIC connection that received the request that resulted in this task starting. If the task was not started in this way, TCPIPJOB returns blanks.

This function depends on Communication Server TCP/IP Network Access Control support being activated and the CLIENTIPADDRESS being configured into a Network Security Zone.

TCPIP SERVICE (*data-area*)

Returns the 8-character name of the TCPIP SERVICE resource associated with the IPIC connection that received the request that resulted in this task starting. If the task was not started in this way, TCPIP SERVICE returns blanks.

TCPIPZONE (*data-area*)

Returns the 8-character name of the TCP/IP network security zone, if any, associated with the IPIC connection that received the request that resulted in this task starting. If there is no TCP/IP network security zone, or the task was not started in this way, TCPIPZONE returns blanks.

TCPIPZONE depends on Communication Server TCP/IP Network Access Control support being activated and the CLIENTIPADDRESS being configured into a Network Security Zone.

TRNGRPID (*data-area*)

Returns, in a 28-byte area, a unique identifier that represents the transaction group ID of the originating transaction.

TRANSACTION(*data-area*)

Returns the 4-character name of the transaction that this task is running.

USERCORRDATA(*data-area*)

Returns, in a 64-byte area, the user correlator data that was added to the associated data origin descriptor by means of an XAPADMGR global user exit program. This field is created when the originating task is started. If the global user exit program is not driven at that point, USERCORRDATA returns blanks.

USERID(*data-area*)

Returns the 8-character user ID associated with this task.

Conditions**INVREQ**

RESP2 values:

2 The command was specified with no arguments.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

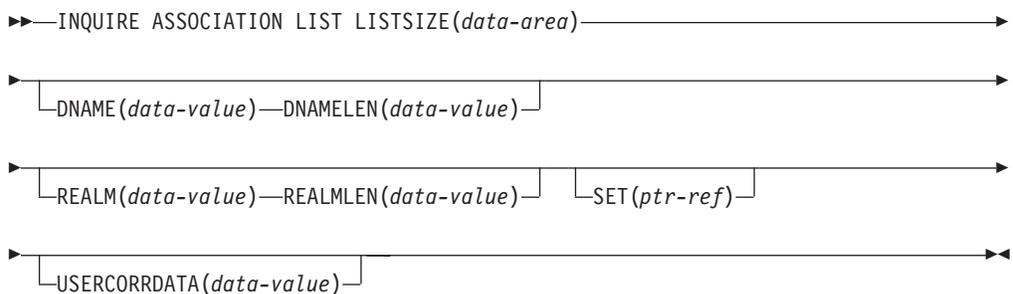
TASKIDERR

RESP2 values:

1 The task specified on the ASSOCIATION option was not found.

INQUIRE ASSOCIATION LIST

The **INQUIRE ASSOCIATION LIST** command returns a list of user tasks that are in the local region.

INQUIRE ASSOCIATION LIST

Conditions: INVREQ, LENGERR, NOTAUTH

This command is threadsafe.

Description

User tasks are tasks that are associated with user-defined transactions or with transactions supplied by CICS. You can restrict the list to tasks that match a number of filters.

You can use **INQUIRE ASSOCIATION LIST** to filter tasks on user correlation data that has been added to the associated data origin descriptors of the tasks by an

XAPADMGR global user exit program. You can also search on certain fields in the origin data portion of the association data to find those tasks and transaction group IDs that share a set of common values. See “Filtering options” for information about the fields that can be filtered.

The command returns, in SET, the address of a list of tasks. Each entry in the list identifies a task that matches the DNAME and REALM, and USERCORRDATA filters. The number of items in the list is returned in LISTSIZE.

Filtering options

The DNAME, REALM, and USERCORRDATA options are three separate filters. The following rules apply:

- If you specify a filter, only the tasks which match the criteria of the filter are returned.
- If you specify more than one filter, the tasks which match both filters are returned.
- If you do not specify a filter, all tasks are returned.

Options

DNAME (*data-value*)

Specifies UTF-8 character field, up to a maximum of 246 characters, including 2 characters for opening and closing parentheses. You must specify parentheses in the DNAME option. DNAME is a filter to return a list of distinguished names for the realm specified in the REALM option. Distinguished names are represented in UTF-8 encoding, therefore null values are represented with ASCII blanks. An empty list is returned if you specify this option and you do not have the correct z/OS release.

The following search forms are accepted:

(attr=value)
(attr=value*)

where:

- attr is the first attribute in the distinguished name for the realm, specified in the REALM option. This attribute is case-sensitive.
- value is the first value in the distinguished name, which can be a generic name if value* is specified. * represents zero or more characters. This attribute is case-sensitive.

For example, if a distinguished name is in the following format:

CN=John Smith

the search argument can be in this format:

(CN=John Smith)

or a generic form can be in this format:

(CN=John S*)

If a generic filter, for example, (CN=*), is specified, only the tasks that have distinguished names with the first attribute specified are included.

If you are filtering on a name that is greater than 244 characters in length, you must use a generic filter.

If value is not specified, or DNAME is not set, all distinguished names for the specified realm are included.

See Filtering options for information about how DNAME operates with REALM and USERCORRDATA.

DNAMELEN(*data-value*)

Specifies the length of the DNAME option. DNAMELEN is a numeric value, up to a maximum of 246.

LISTSIZE(*data-area*)

Returns, as a fullword binary number, the number of items in the list addressed by the SET option. Each entry in the list identifies a task that matches the DNAME and REALM, and USERCORRDATA filters. If one or more of the filters do not match any task, LISTSIZ returns zero.

REALM(*data-value*)

Specifies the realm name in UTF-8 encoding, therefore null values are represented with ASCII blanks. The realm is a component of a distributed identity and defines the region where a security ID applies. If you are using WebSphere Application Server, the realm name can be the service that provides access to the registry where the user is defined. The LDAP server configuration listen statement provides the realm name in URL format.

An empty list is returned if you specify this option and you do not have the correct z/OS release.

If a value is not specified, or REALM is not set, all realms are included.

See Filtering options for information about how REALM operates with DNAME and USERCORRDATA.

REALMLEN(*data-value*)

Specifies the length of the REALM option. REALMLEN is a numeric value, up to a maximum of 255.

SET(*ptr-ref*)

Specifies the address of a list of 4-byte packed-decimal task numbers. Each entry in the list identifies a task that matches the DNAME and REALM, and USERCORRDATA filters. If one or more of the filters do not match any task, the SET pointer contains a null value.

CICS obtains the storage for this list and frees it when the inquiring task issues another **INQUIRE ASSOCIATION LIST** command or ends. The task cannot free the storage.

USERCORRDATA(*data-value*)

Specifies a subset (up to 64 bytes) of the user correlation data added to the associated data origin descriptor by an XAPADMGR global user exit program. This data is used as a filter to return a list of task numbers that match this request.

The filter can contain the following “wildcard” characters:

- ? matches exactly one arbitrary character.
- * matches zero or more arbitrary characters.

See Filtering options for information about how USERCORRDATA operates with DNAME and REALM.

Conditions

INVREQ

RESP2 values:

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

ATOMSERVICE (*data-value*)

Specifies the name of the ATOMSERVICE resource about which you are inquiring. The name can be up to 8 characters in length.

ATOMTYPE (*cvda*)

Returns a CVDA value indicating the type of Atom document that is produced by this ATOMSERVICE definition. CVDA values are as follows:

CATEGORY

An Atom category document, which lists the categories of documents in a collection.

COLLECTION

An Atom collection document, which contains a group of entry documents that can be edited.

FEED An Atom feed document, which describes the metadata for a feed and contains entry documents that provide data for the feed.

SERVICE

An Atom service document, which provides information about the collections of entry documents that are available on the server and can be added to or edited.

UNKNOWN

The ATOMTYPE cannot be determined, this is probably because the associated configuration file cannot be read.

BINDFILE (*data-area*)

Returns a 255-character data area containing the fully qualified (absolute) or relative name of the XML binding specified in this ATOMSERVICE definition. The XML binding is stored in z/OS UNIX System Services, and it specifies the data structures used by the CICS resource that supplies the data for the Atom document that is returned for this resource definition. Service and category documents do not use an XML binding, so, if ATOMTYPE is SERVICE or CATEGORY, BINDFILE returns blanks.

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see FORMATTIME in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

CONFIGFILE (*data-area*)

Returns a 255-character data area containing the fully qualified (absolute) or relative name of the Atom configuration file specified in this ATOMSERVICE definition. The Atom configuration file is stored in z/OS UNIX System Services, and it contains XML that specifies metadata and content for the Atom document that is returned for this resource definition.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

ENABLESTATUS (*cvda*)

Returns a CVDA value indicating the status of this ATOMSERVICE definition. CVDA values are as follows:

ENABLED

The ATOMSERVICE definition is enabled.

DISABLED

The ATOMSERVICE definition is disabled. An ATOMSERVICE definition with this status can be discarded.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

BUNDLE

The resource was installed by a bundle deployment.

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

RESOURCENAME(*data-area*)

Returns the 16-character name of the CICS resource that provides the data for this Atom feed or collection. This option does not apply for an Atom service or category document.

RESOURCETYPE(*cvda*)

Returns a CVDA value indicating the type of CICS resource that provides the data for this Atom feed or collection. This option does not apply for an Atom service or category document. CVDA values are as follows:

FILE A CICS file.

PROGRAM

A service routine, which is a CICS application program written to supply content for Atom entries.

TSQUEUE

A temporary storage queue.

NOTAPPLIC

If the value of ATOMTYPE is SERVICE or CATEGORY, the resource type is not applicable.

URIMAP(*data-area*)

Returns the 8-character URIMAP name that indicates the URI associated with this ATOMSERVICE definition. If there is no auto-generated URIMAP associated with this ATOMSERVICE definition, this field is empty.

XMLTRANSFORM(*data-area*)

Returns the 32-character name of the XMLTRANSFORM resource associated with the ATOMSERVICE definition. If the value of ATOMTYPE is SERVICE or CATEGORY, this field is empty.

Conditions

END

RESP2 values:

2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

|
|
|
|
|
|
|
|
|

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

NOTFND

RESP2 values:

- 3 The ATOMSERVICE cannot be found.

INQUIRE AUTINSTMODEL

Find out whether an autoinstall model is installed.

INQUIRE AUTINSTMODEL

►►—INQUIRE AUTINSTMODEL(*data-value*)—◄◄

Conditions: END, ILLOGIC, MODELIDERR, NOTAUTH

Description

The INQUIRE AUTINSTMODEL command allows you to determine whether a particular autoinstall model is installed (defined in the current execution of your CICS system).

Browsing

You can also browse through all of the autoinstall models installed in your system by using the browse options (START, NEXT, and END) on INQUIRE AUTOINSTALL commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

Options

AUTINSTMODEL (*data-value*)

specifies the 8-character identifier of the autoinstall model about which you are inquiring.

Conditions

END

RESP2 values:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

MODELIDERR

RESP2 values:

- 1 The model specified cannot be found.

NOTAUTH

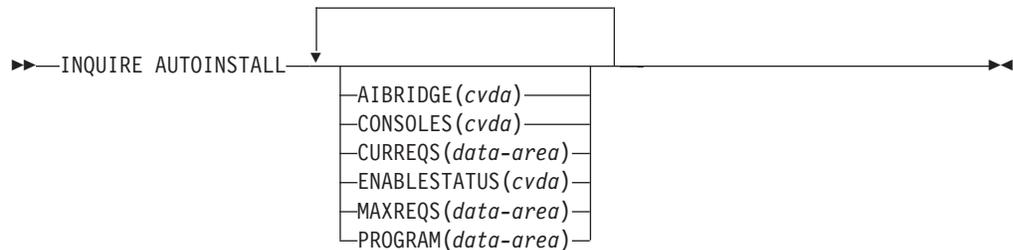
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

INQUIRE AUTOINSTALL

Retrieve autoinstall values.

INQUIRE AUTOINSTALL



Conditions: NOTAUTH

Description

The INQUIRE AUTOINSTALL returns information relating to the automatic installation (autoinstall) of z/OS Communications Server terminals, APPC sessions, virtual terminals (bridge facilities) used by the 3270 bridge mechanism, and MVS consoles in your CICS system.

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Options

AIBRIDGE (cvda)

returns a CVDA value indicating whether the autoinstall user replaceable program (URM) is called for bridge facilities. The CVDA values are:

AUTOTERMID

Bridge facilities are defined automatically by CICS. The autoinstall user replaceable program is not called.

URMTERMID

The autoinstall user replaceable program is called.

CONSOLES (cvda)

returns a CVDA value indicating the status of console autoinstall in CICS. The CVDA values are:

PROGAUTO

Consoles can be autoinstalled if ENABLESTATUS returns a CVDA of ENABLED. The autoinstall control program is called for the install and delete functions.

FULLAUTO

Consoles can be autoinstalled if ENABLESTATUS returns a CVDA of ENABLED. The autoinstall control program is not called for the install

and delete functions, and CICS generates the terminal identifier automatically for the consoles it autoinstalls.

NOAUTO

Consoles cannot be autoinstalled.

CURREQS (*data-area*)

returns a fullword binary field indicating the number of terminal autoinstall requests that are currently being processed. This count does not include terminals already installed in this manner.

ENABLESTATUS (*cvda*)

returns a CVDA value indicating the overall status of the CICS autoinstall facility. CVDA values are:

DISABLED

Neither consoles nor terminals can be autoinstalled in CICS. DISABLED is returned for the following conditions:

Terminals

MAXREQS equal 0, or the autoinstall control program is disabled.

Consoles

1. CONSOLES CVDA returns NOAUTO.
2. CONSOLES CVDA returns PROGAUTO but autoinstall control program is disabled.

ENABLED

Either consoles or terminals or both can be autoinstalled in CICS. If you want to check whether ENABLED applies to consoles, terminals, or both, check the values returned on other options. ENABLED is returned for the following conditions:

Terminals

MAXREQS not equal 0 and autoinstall control program is enabled.

Consoles

1. CONSOLES CVDA returns FULLAUTO.
2. CONSOLES CVDA returns PROGAUTO and autoinstall control program is enabled.

MAXREQS (*data-area*)

returns a fullword binary field indicating the largest number of autoinstall requests that can be processed concurrently. Note that this value has no effect on the total number of terminals that can be installed automatically. (The MAXREQS option corresponds to the AIQMAX system initialization parameter.)

PROGRAM (*data-area*)

returns the 8-character name of the installation-supplied program used in the autoinstall process. This is either the CICS-supplied default autoinstall program, DFHZATDX, or a user-written program.

Conditions

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

INQUIRE BEAN

Retrieve information about an installed enterprise bean.

INQUIRE BEAN

```
▶▶—INQUIRE BEAN(data-value)—┬──CORBASERVER(data-area)──┬──DJAR(data-area)──▶▶
```

Conditions: END, INVREQ, NOTAUTH , NOTFND

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

INQUIRE BEAN retrieves the CorbaServer and DJAR names associated with an installed enterprise bean.

Browsing

You can also browse through all of the enterprise beans installed in your system by using the browse options (START, NEXT, and END) on **INQUIRE BEAN** commands.

See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

Options

BEAN(*data-value*)

specifies the name of the enterprise bean.

CORBASERVER(*data-area*)

specifies a 4-character area to receive the name of the associated CorbaServer.

DJAR(*data-area*)

specifies a 32-character area to receive the name of the deployed JAR file that contains the bean.

Conditions

END

RESP2 values:

2 There are no more beans to browse.

INVREQ

RESP2 values:

24 CorbaServer not specified.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

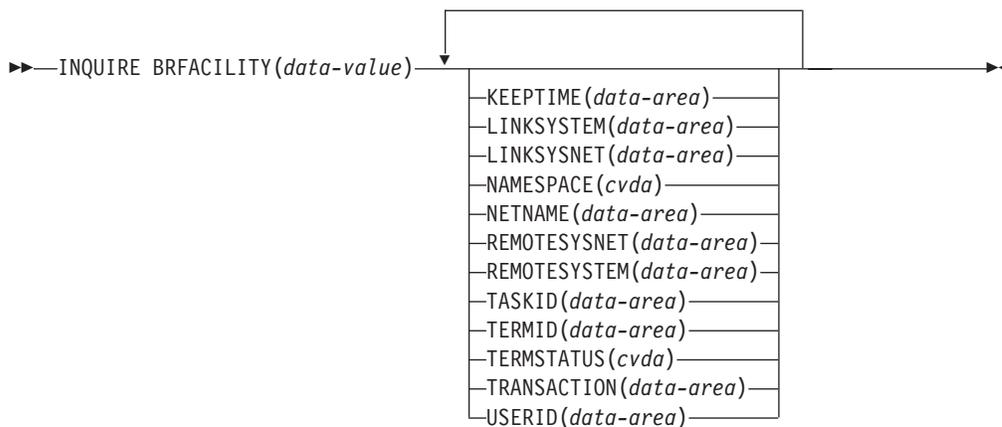
NOTFND

RESP2 values:

INQUIRE BRFACILITY

Retrieve information about a virtual terminal (bridge facility) used by the 3270 bridge mechanism.

INQUIRE BRFACILITY



Conditions: END, ILLOGIC, NOTAUTH, NOTFOUND

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The INQUIRE BRFACILITY command returns information about a bridge facility. This is a virtual terminal used by the 3270 bridge mechanism to simulate a real 3270 when running a CICS 3270 application in a bridged environment. You can use this command in any application running in the Link3270 bridge program or AOR region where the bridge facility was created, to retrieve information about any active bridge facility, even if it is not your principal facility.

Browsing

You can also browse through the bridge facilities installed in your system by using the browse options (START, NEXT, and END) on INQUIRE BRFACILITY commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

Options

BRFACILITY(*data-value*)

Specifies the 8-byte facility token of the bridge facility about which you are inquiring.

KEEPTIME(*data-area*)

Returns a full word binary field showing the length of time (in seconds) that the bridge facility is kept if inactive.

- If the bridge facility being displayed is a Link-Bridge, this value is the time that was specified when the facility was allocated, or a default value of 5

minutes. If the KEEPTIME value is larger than the value of the **BRMAXKEEPTIME** system initialization parameter, it is reduced to **BRMAXKEEPTIME**.

- If the facility being displayed is a Web-Bridge, the KEEPTIME value is initially set to be the Webdelay terminal keep time (the second part of the **WEBDELAY** system initialization parameter).

LINKSYSNET (*data-area*)

Returns the 8-byte applid of the AOR if the Link3270 bridge request is routed to another region. If the request is processed in the same region as the Link3270 bridge program, then this field is blank. This field may change if dynamic transaction routing makes more than one attempt at running the first transaction in a Link3270 session. This field is only set in the Link3270 bridge program region.

LINKSYSTEM (*data-area*)

Returns the 4-byte SYSID of the AOR if the Link3270 bridge request is routed to another region. If the request is processed in the same region as the Link3270 bridge program, then this field is blank. This field may change if dynamic transaction routing makes more than one attempt at running the first transaction in a Link3270 session. This field is only set in the Link3270 bridge program region.

NAMESPACE (*cvda*)

Returns a CVDA value indicating the scope of the name space used to allocate bridge facility names. CVDA values are:

LOCAL

The bridge facility was allocated by the START BREXIT bridge mechanism, so its name is unique only in the local region where it is created.

SHARED

The bridge facility was allocated by the Link3270 bridge mechanism, so its name is unique across all CICS Link3270 bridge regions in the CICSplex who have access to a shared DFHBRNSF namespace file.

NETNAME (*data-area*)

Specifies the 8-byte virtual netname name of the bridge facility about which you are inquiring.

REMOTESYSNET (*data-area*)

Returns an 8-byte field giving the applid of the router. This field is only set in the AOR region. It is blank if the AOR is the router region.

REMOTESYSTEM (*data-area*)

Returns a 4-byte giving the SYSID of the router. This field is only set in the AOR region. It is blank if the AOR is the router region.

TASKID (*data-area*)

Returns a full word binary field showing the number of the task running the user transaction. This field is only set in the AOR. This field is zero the bridge facility is currently not in use.

TERMID (*data-area*)

Specifies the 4-byte virtual terminal name of the bridge facility about which you are inquiring.

TERMSTATUS (*cvda*)

Returns a CVDA value indicating the status of the bridge facility. CVDA values are:

ACQUIRED

The bridge facility is currently in use.

AVAILABLE

The bridge facility is not in use. It can be reused by the client.

RELEASED

SET BRFACILITY RELEASED has been issued for the bridge facility. It will be deleted on the next cleanup cycle.

TRANSACTION(*data-area*)

Returns a 4-byte field giving the name of the user transaction being run by the 3270 bridge, as known in the current region. This value is blank if the bridge facility is currently not in use.

USERID(*data-area*)

Returns an 8-byte field giving the userid associated with this bridge facility.

Conditions**END**

RESP2 values:

- 1 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

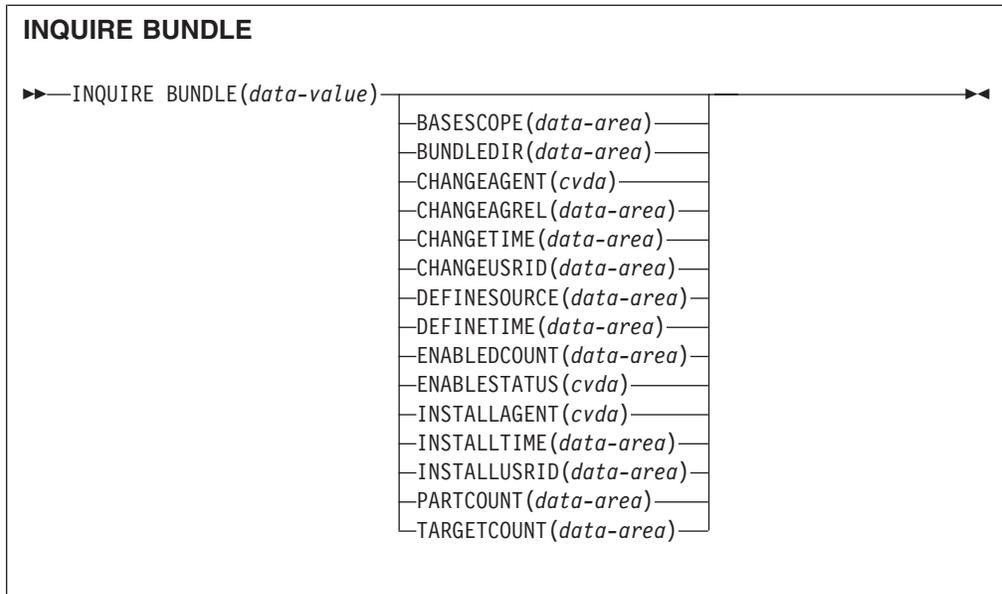
NOTFOUND

RESP2 values:

- 1 The specified bridge facility cannot be found.

INQUIRE BUNDLE

Retrieve information about an installed BUNDLE.



Conditions: END, ILLOGIC, INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the **INQUIRE BUNDLE** command to retrieve information about an installed BUNDLE resource. The contents of a BUNDLE resource are defined in a manifest that can contain imports, exports, and definitions. The definitions section of the manifest describe the resources that CICS dynamically creates for you when you install the BUNDLE resource. Use this command to find out the location of the bundle on z/OS UNIX, the number of imports, exports, and definitions that are listed in the manifest, and how many of those definitions are currently enabled in the CICS region.

For more detailed information about each import, export, and definition, use the **INQUIRE BUNDLEPART** command.

Browsing

You can browse through all the BUNDLE resources that are installed in your region by using the browse options, START, NEXT, and END, on **INQUIRE BUNDLE** commands.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

BASESCOPE(*data-area*)

Returns the 1 - 255 character string that defines the root namespace for the contents of the bundle.

BUNDLE(*data-value*)

Specifies the name of the BUNDLE resource about which you are inquiring. The name can be up to 8 characters in length.

BUNDLEDIR(*data-area*)

Returns the 1 - 255 character fully qualified name of the root directory for the bundle on z/OS UNIX.

CHANGEAGENT(*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL(*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID(*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE(*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT** value. For more information, see **Summary of the resource signature field values in the Resource Definition Guide**.

DEFINETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

ENABLEDCOUNT(*data-area*)

Returns the current number of resources that were dynamically created by the bundle and are enabled in the CICS region.

ENABLESTATUS(*cvda*)

Returns the status of the BUNDLE:

ENABLED

The BUNDLE is ready for use.

DISABLED

The BUNDLE is not available.

ENABLING

The BUNDLE is being initialized. It is creating and enabling the resources that are defined in the bundle manifest file.

DISABLING

The BUNDLE is quiescing before entering DISABLED state. It disables any resources that it enabled.

DISCARDING

A DISCARD command has been issued for the BUNDLE. The BUNDLE is quiescing before being discarded. The bundle discards any resources that it disabled.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID (*data-area*)

Returns the 8-character user ID that installed the resource.

PARTCOUNT (*data-area*)

Returns the total number of imports, exports, and definition statements that are defined in the bundle manifest.

TARGETCOUNT (*data-area*)

Returns the total number of dynamically created resources in the bundle. CICS automatically enables the BUNDLE resource when all of the dynamically created resources are in an enabled state.

Conditions**END**

RESP2 values:

2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

1 A START command has been issued when a browse of BUNDLE resources is already in progress, or a NEXT or an END command has been issued when a browse of BUNDLE resources is not in progress.

INVREQ

RESP2 values:

7 CICS failed to link to the registered bundle callback program.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

NOTFND

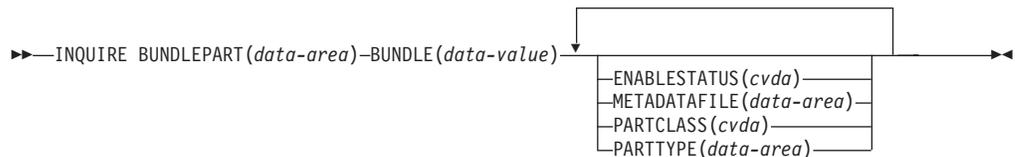
RESP2 values:

- 3 The BUNDLE cannot be found.

INQUIRE BUNDLEPART

Retrieve information about the resources that are contained in an installed BUNDLE resource.

INQUIRE BUNDLEPART



Conditions: END, ILLOGIC, INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the **INQUIRE BUNDLEPART** command to return information about the resources that are contained in an installed BUNDLE resource. You can use the **INQUIRE BUNDLEPART** command only in browse mode.

Browsing

You can browse through all the resources that are installed in your region by a BUNDLE resource by using the browse options (START, NEXT, and END) on the **INQUIRE BUNDLEPART** command.

Options

BUNDLE(data-value)

Specify the 1 - 8 character name of the BUNDLE resource that you want to browse when using the START option.

BUNDLEPART(data-area)

Returns the 1 - 255 character name of a resource that is contained in the bundle.

ENABLESTATUS(cvda)

Returns the status of the resource in the bundle:

ENABLED

The resource is ready for use.

DISABLED

The resource is not available.

ENABLING

The resource is being created as part of the BUNDLE resource installation.

DISABLING

The resource is being disabled.

DISCARDING

A DISCARD command has been issued for the BUNDLE resource. The resource in the bundle is disabled and is being discarded.

UNUSABLE

The resource is in an unusable state.

METADATAFILE(*data-area*)

Returns the 1 - 255 character name of the file on z/OS UNIX that describes the resource, as defined in the manifest. The value is an absolute path from the root of the bundle directory.

PARTCLASS(*cvda*)

Returns the class of the resource that is defined in the manifest. The following values are valid:

DEFINITION

The resource is defined as a definition in the manifest.

EXPORT

The resource is defined as an export in the manifest. The resource is available to other services installed in the CICS region.

IMPORT

The resource is defined as an import in the manifest. The resource is required by the application in the CICS region.

PARTTYPE(*data-area*)

Returns the 1 - 255 character resource type as a URI. For example, the XMLTRANSFORM resource has the URI `http://www.ibm.com/xmlns/prod/cics/bundle/XMLTRANSFORM`.

Conditions**END**

RESP2 values:

2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

1 A START command has been issued when a browse of BUNDLE resources is already in progress, or a NEXT or an END command has been issued when a browse of BUNDLE resources is not in progress.

INVREQ

RESP2 values:

8 A BUNDLE name must be specified on **START BUNDLEPART**.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

101 The user associated with the issuing task is not authorized to access the BUNDLE resource in the way required by this command.

NOTFND

RESP2 values:

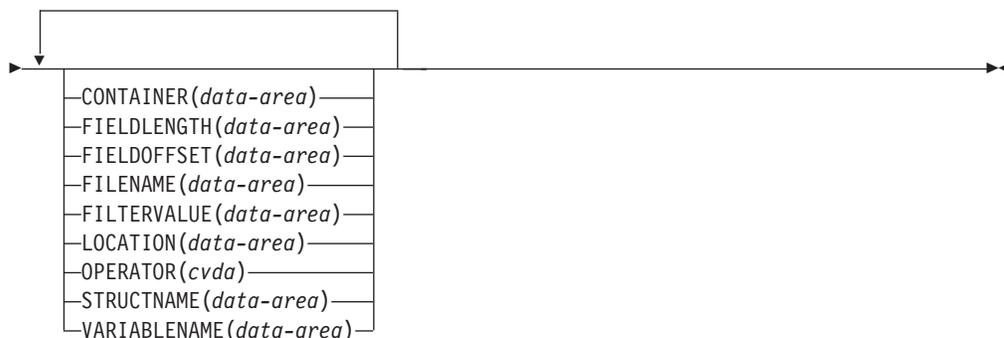
3 The BUNDLE cannot be found.

INQUIRE CAPDATAPRED

Retrieve information about an application data predicate that is defined for a capture specification.

INQUIRE CAPDATAPRED

►►—INQUIRE CAPDATAPRED—CAPTURESPEC(*data-value*)—EVENTBINDING(*data-value*)—►►



Conditions: END, ILLOGIC, INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

The INQUIRE CAPDATAPRED command returns details of each application data predicate that is defined for a capture specification. The INQUIRE CAPDATAPRED command can be used only in browse mode.

Browsing

You can browse through all of the application data predicates for a capture specification by using the browse options (START, NEXT, and END) on the **INQUIRE CAPDATAPRED** command.

Options

CONTAINER(*data-area*)

Specifies a 16-character data area to receive the name of the container that contains the data when the value of the LOCATION option is equal to CHANNEL or FROMCHANNEL; otherwise, this option contains blanks. Values that are less than 16 characters are padded with blanks.

| **FIELDLENGTH**(*data-area*)

| Returns a fullword binary field that contains the length, in bytes, of the data to
| be tested by this predicate. The value of the LOCATION option identifies the
| data source.

| **FIELDOFFSET**(*data-area*)

| Returns a fullword binary field that contains the offset into the data source,
| which is indicated by the value of the LOCATION option, that contains the data
| to be tested by this predicate.

| **FILENAME**(*data-area*)

| Specifies a 32-character data area to receive the first 32 characters of the name
| of the file that contains the imported language structure that defines this
| predicate. This option contains all blanks if an imported language structure
| was not used. Values that are less than 32 characters are padded with blanks.
| Data that is returned as a result of this option is encoded in your local coded
| character set identifier (CCSID).

| **FILTERVALUE**(*data-area*)

| Specifies a 255-character data area to receive the value of the application data
| predicate. Predicates that are not characters are converted to their character
| representation. Values that are less than 255 characters are padded with blanks.
| Data that is returned as a result of this option is encoded in your local coded
| character set identifier (CCSID).

| **LOCATION**(*data-area*)

| Specifies a 32-character data area to receive the location of the data to be
| tested. For the CICS event binding editor, this value is the same as the value
| for the Location field in the Variable location and format section of the
| Application Data predicate dialog when you add or edit application data
| predicates for an application event. Values that are less than 32 characters are
| padded with blanks.

| **OPERATOR**(*cvda*)

| Returns a CVDA that defines the operator that is used with the value of the
| FILTERVALUE option to evaluate the predicate. Possible CVDA values are as
| follows:

| **DOESNOTEQUAL**

| The predicate evaluates true when the value of the data item that is
| defined by the LOCATION, FIELDOFFSET, and FIELDLENGTH values is not
| equal to the value of the FILTERVALUE option.

| **DOESNOTEXIST**

| The predicate evaluates true when the data source that is specified for
| the LOCATION option does not exist.

| **DOESNOTSTART**

| The predicate evaluates true when the value of the data item that is
| defined by the LOCATION, FIELDOFFSET, and FIELDLENGTH values does
| not start with the value of the FILTERVALUE option.

| **EQUALS**

| The predicate evaluates true when the value of the data item that is
| defined by the LOCATION, FIELDOFFSET, and FIELDLENGTH values is equal
| to the value of the FILTERVALUE option.

| **EXISTS**

| The predicate evaluates true when the data source that is specified for
| the LOCATION option exists.

| **GREATERTHAN**

| The predicate evaluates true when the value of the data item that is
| defined by the LOCATION, FIELDOFFSET, and FIELDLENGTH values is
| greater than the value of the FILTERVALUE option.

| **ISNOTGREATER**

| The predicate evaluates true when the value of the data item that is
| defined by the LOCATION, FIELDOFFSET, and FIELDLENGTH values is equal
| to or less than the value of the FILTERVALUE option.

| **ISNOTLESS**

| The predicate evaluates true when the value of the data item that is
| defined by the LOCATION, FIELDOFFSET, and FIELDLENGTH values is equal
| to or greater than the value of the FILTERVALUE option.

| **LESSTHAN**

| The predicate evaluates true when the value of the data item that is
| defined by the LOCATION, FIELDOFFSET, and FIELDLENGTH values is less
| than the value of the FILTERVALUE option.

| **STARTSWITH**

| The predicate evaluates true when the value of the data item that is
| defined by the LOCATION, FIELDOFFSET, and FIELDLENGTH values starts
| with the value of the FILTERVALUE option.

| **STRUCTNAME** (*data-area*)

| Specifies a 32-character data area to receive the first 32 characters of the name
| of the imported language structure used to define this predicate. This field
| contains all blanks when an imported language structure is not used. Values
| that are less than 32 characters are padded with blanks. Data that is returned
| as a result of this option is encoded in your local coded character set identifier
| (CCSID).

| **VARIABLENAME** (*data-area*)

| Specifies a 32-character data area to receive the first 32 characters of the name
| of the variable in the imported language structure used to define this
| predicate. This field contains all blanks when an imported language structure
| is not used. Values that are less than 32 characters are padded with blanks.
| Data that is returned as a result of this option is encoded in your local coded
| character set identifier (CCSID).

| **Conditions**

| **END**

| RESP2 values:

- | 2 There are no more resource definitions of this type.
- | 8 The event binding has been deleted so browse has been terminated
| early.

| **ILLOGIC**

| RESP2 values:

- | 1 You have issued a START command when a browse of this resource
| type is already in progress, or you have issued a NEXT or an END
| command when a browse of this resource type is not in progress.

| **INVREQ**

- | 4 An EVENTBINDING name has not been specified for the START
| CAPDATAPRED browse.

5 A CAPTURESPEC name has not been specified for the START CAPDATAPRED browse.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

101 The user associated with the issuing task is not authorized to read the event binding.

NOTFND

RESP2 values:

2 The specified capture specification cannot be found.

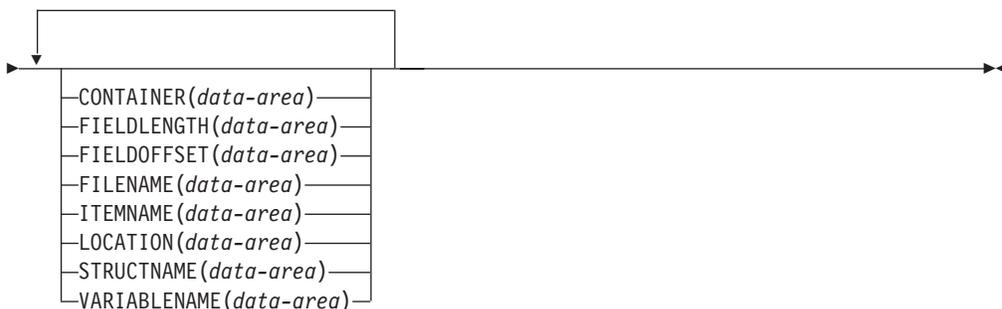
3 The specified event binding cannot be found.

INQUIRE CAPINFOSRCE

Retrieve information about an information source that is defined for a capture specification.

INQUIRE CAPINFOSRCE

►►—INQUIRE CAPINFOSRCE—CAPTURESPEC(*data-value*)—EVENTBINDING(*data-value*)—►►



Conditions: END, ILLOGIC, INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

The INQUIRE CAPINFOSRCE command returns details of each information source that is defined for a capture specification. The INQUIRE CAPINFOSRCE command can be used only in browse mode.

Browsing

You can browse through all of the information sources for a capture specification by using the browse options (START, NEXT, and END) on the **INQUIRE CAPINFOSRCE** command.

Options

CONTAINER(*data-area*)

Specifies a 16-character data area to receive the name of the container that contains the data when the value of the LOCATION option is equal to CHANNEL or FROMCHANNEL; otherwise, this option contains blanks. Values that are less than 16 characters are padded with blanks.

FIELDLENGTH(*data-area*)

Returns a fullword binary field that contains the length, in bytes, of the application data to be captured from the information source which is indicated by the value of the LOCATION option. The FIELDLENGTH option is 0 for all application event context and command option capture items, and for all system event capture items.

FIELDOFFSET(*data-area*)

Returns a fullword binary field that contains the offset into the application data source, which is indicated by the value of the LOCATION option, that contains the data to be captured by the capture specification. The FIELDOFFSET option is 0 for all application event context and command option capture items, and for all system event capture items.

FILENAME(*data-area*)

Specifies a 32-character data area to receive the first 32 characters of the name of the file that contains the imported language structure that defines this information source. This option contains all blanks for a system event or when an imported language structure was not used to define an application event. Values that are less than 32 characters are padded with blanks. The returned value is encoded in CCSID for the local region.

ITEMNAME(*data-area*)

Specifies a 32-character data area to receive the name that was specified for this item of emitted business information. Values that are less than 32 characters are padded with blanks.

LOCATION(*data-area*)

Specifies a 32-character data area to receive the name of the information source. When you use the CICS event binding editor, this value is the same as the value you select from the list of available data when you add or edit an information source for an application or system event. Values that are less than 32 characters are padded with blanks.

STRUCTNAME(*data-area*)

Specifies a 32-character data area to receive the first 32 characters of the name of the imported language structure used to define this information source. This field contains all blanks for a system event or when an imported language structure is not used to define an application event. Values that are less than 32 characters are padded with blanks. The returned value is encoded in CCSID for the local region.

VARIABLENAME(*data-area*)

Specifies a 32-character data area to receive the first 32 characters of the name of the variable in the imported language structure used to define this information source. This field contains all blanks for a system event or when an imported language structure is not used to define an application event. Values that are less than 32 characters are padded with blanks. The returned value is encoded in CCSID for the local region.

Conditions

END

RESP2 values:

- 2 There are no more resource definitions of this type.
- 8 The event binding has been deleted so browse has been terminated early.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

INVREQ

- 4 An EVENTBINDING name has not been specified for the START CAPINFOSRCE browse.
- 5 A CAPTURESPEC name has not been specified for the START CAPINFOSRCE browse.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to read the event binding.

NOTFND

RESP2 values:

- 2 The specified capture specification cannot be found.
- 3 The specified event binding cannot be found.

INQUIRE CAOPTPRED

Retrieve information about a capture option predicate that is defined for a capture specification.

INQUIRE CAOPTPRED

►►—INQUIRE CAOPTPRED—CAPTURESPEC(*data-value*)—EVENTBINDING(*data-value*)—►►



Conditions: END, ILLOGIC, INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

The INQUIRE CAPOPTPRED command returns details of each capture option predicate, including the primary predicate, that is defined for a capture specification. The INQUIRE CAPOPTPRED command can be used only in browse mode.

Browsing

You can browse through all of the capture option predicates for a capture specification by using the browse options (START, NEXT, and END) on the **INQUIRE CAPOPTPRED** command.

Options

FILTERVALUE (*data-area*)

Specifies a 255-character data area to receive the value of the application or system event option named in the OPTIONNAME option. Values that are less than 255 characters are padded with blanks.

OPERATOR (*cvda*)

Returns a CVDA that defines the operator that is used with the value of the FILTERVALUE option to evaluate the predicate. Possible CVDA values are as follows:

DOESNOTEQUAL

The predicate evaluates true when the value of the OPTIONNAME option is not equal to the value of the FILTERVALUE option.

DOESNOTEXIST

The predicate evaluates true when the OPTIONNAME option is not specified on the EXEC CICS command.

DOESNOTSTART

The predicate evaluates true when the value of the OPTIONNAME option does not start with the value of the FILTERVALUE option.

EQUALS

The predicate evaluates true when the value of the OPTIONNAME option is equal to the value of the FILTERVALUE option.

EXISTS

The predicate evaluates true when the OPTIONNAME option is specified on the EXEC CICS command.

GOHIGHERTHAN

The predicate evaluates true when the value of the threshold option named in OPTIONNAME crosses above the threshold percentage returned in FILTERVALUE.

GOLOWERTHAN

The predicate evaluates true when the value of the threshold option named in OPTIONNAME crosses below the threshold percentage returned in FILTERVALUE.

GREATERTHAN

The predicate evaluates true when the value of the OPTIONNAME option is greater than the value of the FILTERVALUE option.

|

| **ISNOTGREATER**

| The predicate evaluates true when the value of the OPTIONNAME option
| is equal to or less than the value of the FILTERVALUE option.

|

| **ISNOTLESS**

| The predicate evaluates true when the value of the OPTIONNAME option
| is equal to or greater than the value of the FILTERVALUE option.

|

| **LESSTHAN**

| The predicate evaluates true when the value of the OPTIONNAME option
| is less than the value of the FILTERVALUE option.

|

| **STARTSWITH**

| The predicate evaluates true when the value of the OPTIONNAME option
| starts with FILTERVALUE.

|

| **OPTIONNAME** (*data-area*)

| Specifies a 32-character data area to receive the name of the capture option
| predicate that is specified in the event specification. In the CICS event binding
| editor, this value matches one of the capture options for the capture point.
| Values that are less than 32 characters are padded with blanks.

|

| **Conditions**

|

| **END**

| RESP2 values:

- |
- | 2 There are no more resource definitions of this type.
 - | 8 The event binding has been deleted so browse has been terminated
| early.

|

| **ILLOGIC**

| RESP2 values:

- |
- | 1 You have issued a START command when a browse of this resource
| type is already in progress, or you have issued a NEXT or an END
| command when a browse of this resource type is not in progress.

|

| **INVREQ**

- |
- | 4 An EVENTBINDING name has not been specified for the START
| CAPOPTPRED browse.
 - | 5 A CAPTURESPEC name has not been specified for the START
| CAPOPTPRED browse.

|

| **NOTAUTH**

| RESP2 values:

- |
- | 100 The user associated with the issuing task is not authorized to use this
| command.
 - | 101 The user associated with the issuing task is not authorized to read the
| event binding.

|

| **NOTFND**

| RESP2 values:

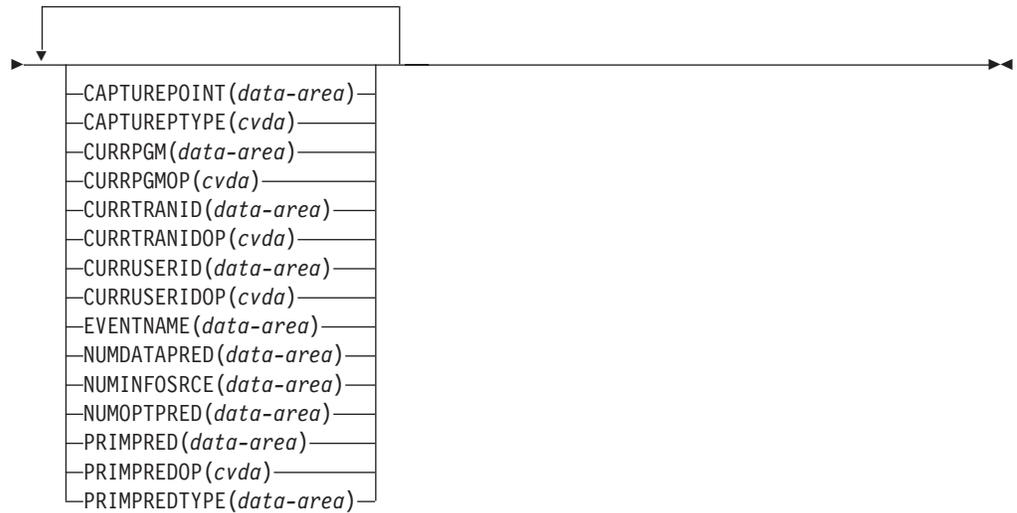
- |
- | 2 The specified capture specification cannot be found.
 - | 3 The specified event binding cannot be found.
- |

INQUIRE CAPTURESPEC

Retrieve information about a capture specification.

INQUIRE CAPTURESPEC

►►—INQUIRE CAPTURESPEC(*data-area*)—EVENTBINDING(*data-value*)—►►



Conditions: END, ILLOGIC, INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

The INQUIRE CAPTURESPEC command returns the attributes associated with a capture specification.

Browsing

You can browse through all the deployed capture specifications installed in the specified event binding using the browse options (START, NEXT, and END) on **INQUIRE CAPTURESPEC** commands.

Options

CAPTUREPOINT(*data-area*)

Specifies a 25-character data area to receive the capture point associated with the capture specification. Its contents will match one of the capture point entries in the event binding tooling. Capture point entries consisting of two words or more are separated with an underscore; for example, LINK_PROGRAM or PROGRAM_INITIATION.

CAPTUREEPTYPE(*cvda*)

Specifies a fullword binary data area to receive a CVDA value identifying the type of capture point. Possible CVDA values are as follows:

PRECOMMAND

Capture point is at the start of a CICS API command.

POSTCOMMAND

Capture point is on completion of a CICS API command.

PROGRAMINIT

Capture point is at program initiation.

SYSTEM

Capture point is a system event.

CAPTURESPEC (*data-area*)

On the non-browse form of this command, specifies the name (1-32 characters) of the capture specification. On the browse form of this command, specifies a 32-character data area to receive the name of the capture specification.

CURRPGM (*data-area*)

Specifies an 8-character data area to receive the value specified by the application context predicate for the current program name. Blanks are returned if no application context predicate for the current program name is defined for this capture specification.

CURRPGMOP (*cvda*)

Returns a CVDA value that defines the operator that is used, together with the value in the CURRPGM option, to evaluate the application context predicate on the current program name. Possible CVDA values are as follows:

ALLVALUES

The predicate always evaluates true; that is, there is no filtering based on the name of the current program.

DOESNOTEQUAL

The predicate evaluates true when the name of the current program is not equal to the value of the CURRPGM option.

DOESNOTSTART

The predicate evaluates true when the name of the current program does not start with the value of the CURRPGM option.

EQUALS

The predicate evaluates true when the name of the current program is equal to the value of the CURRPGM option.

GREATERTHAN

The predicate evaluates true when the name of the current program is greater than the value of the CURRPGM option.

ISNOTGREATER

The predicate evaluates true when the name of the current program is equal to or less than the value of the CURRPGM option.

ISNOTLESS

The predicate evaluates true when the name of the current program is equal to or greater than the value of the CURRPGM option.

LESSTHAN

The predicate evaluates true when the name of the current program is less than the value of the CURRPGM option.

STARTSWITH

The predicate evaluates true when the name of the current program starts with the value of the CURRPGM option.

| **CURRTRANID** (*data-area*)

| Specifies a 4-character data area to receive the value specified by the
| application context predicate for the current transaction name.

| **CURRTRANIDOP** (*cvda*)

| Returns a CVDA value that defines the operator that is used, together with the
| value in the CURRTRANID option, to evaluate the application context predicate on
| the current transaction name. Possible CVDA values are as follows:

| **ALLVALUES**

| The predicate always evaluates true; that is, there is no filtering based
| on the name of the current transaction.

| **DOESNOTEQUAL**

| The predicate evaluates true when the name of the transaction that is
| running is not equal to the value of the CURRTRANID option.

| **DOESNOTSTART**

| The predicate evaluates true when the name of the transaction that is
| running does not start with the value of the CURRTRANID option.

| **EQUALS**

| The predicate evaluates true when the name of the current transaction
| is equal to the value of the CURRTRANID option.

| **GREATERTHAN**

| The predicate evaluates true when the name of the current transaction
| is greater (that is, higher in the collating sequence of possible
| transaction IDs) than the value of the CURRTRANID option.

| **ISNOTGREATER**

| The predicate evaluates true when the name of the current transaction
| is equal to or less (that is, lower in the collating sequence of possible
| transaction IDs) than the value of the CURRTRANID option.

| **ISNOTLESS**

| The predicate evaluates true when the name of the current transaction
| is equal to or greater (that is, higher in the collating sequence of
| possible transaction IDs) than the value of the CURRTRANID option.

| **LESSTHAN**

| The predicate evaluates true when the name of the current transaction
| is less (that is, lower in the collating sequence of possible transaction
| IDs) than the value of the CURRTRANID option.

| **STARTSWITH**

| The predicate evaluates true when the name of the current transaction
| starts with the value of the CURRTRANID option.

| **CURRUSERID** (*data-area*)

| Specifies an 8-character data area to receive the value specified by the
| application context predicate for the user ID that is associated with the current
| transaction.

| **CURRUSERIDOP** (*cvda*)

| Returns a CVDA value that defines the operator that is used, together with the
| value in the CURRUSERID option, to evaluate the application context predicate on
| the user ID. Possible CVDA values are as follows:

| **ALLVALUES**

| The predicate always evaluates true; that is, there is no filtering based
| on the user ID.

|
| **DOESNOTEQUAL**

| The predicate evaluates true when the user ID of the current user is
| not equal to the value of the CURRUSERID option.

| **DOESNOTSTART**

| The predicate evaluates true when the user ID of the current user does
| not start with the value of the CURRUSERID option.

| **EQUALS**

| The predicate evaluates true when the user ID of the current user is
| equal to the value of the CURRUSERID option.

| **GREATERTHAN**

| The predicate evaluates true when the user ID of the current user is
| greater (that is, higher in the collating sequence of possible user IDs)
| than the value of the CURRUSERID option.

| **ISNOTGREATER**

| The predicate evaluates true when the user ID of the current user is
| equal to or less (that is, lower in the collating sequence of possible user
| IDs) than the value of the CURRUSERID option.

| **ISNOTLESS**

| The predicate evaluates true when the user ID of the current user is
| equal to or greater (that is, higher in the collating sequence of possible
| user IDs) than the value of the CURRUSERID option.

| **LESSTHAN**

| The predicate evaluates true when the user ID of the current user is
| less (that is, lower in the collating sequence of possible user IDs) than
| the value of the CURRUSERID option.

| **STARTSWITH**

| The predicate evaluates true when the user ID of the current user starts
| with the value of the CURRUSERID option.

| **EVENTBINDING**(*data-value*)

| Specifies the name (1-32 characters) of the associated event binding.

| **EVENTNAME**(*data-area*)

| Specifies a 32-character data area to receive the associated business event
| name.

| **NUMDATAPRED**(*data-area*)

| Specifies a fullword binary field that is set to the number of application data
| predicates that are defined for this capture specification.

| **NUMINFOSRCE**(*data-area*)

| Specifies a fullword binary field that is set to the number of information
| sources that are defined for this capture specification.

| **NUMOPTPRED**(*data-area*)

| Specifies a fullword binary field that is set to the number of application
| command option or system event option predicates that are defined for this
| capture specification. The total number of predicates includes the primary
| predicate.

| **PRIMPRED**(*data-area*)

| Specifies a 32-character data area to receive the value of the primary predicate
| for this capture specification. The primary predicate for a capture specification
| is the predicate to specify with the EQUALS operator; it helps to avoid a

| performance impact as more capture specifications are added for a particular
| capture point. Blanks are returned if there is no named primary predicate
| defined for this capture point.

| **PRIMPREDOP** (*cvda*)

| Returns a CVDA value that defines the operator that is used, together with the
| value in the PRIMPRED option, to evaluate the primary predicate. Possible
| CVDA values are as follows:

| **ALLVALUES**

| The predicate always evaluates true; that is, there is no filtering based
| on the name of the resource for the command.

| **DOESNOTEQUAL**

| The predicate evaluates true when the resource that is specified by the
| command is not equal to the value of the PRIMPRED option.

| **DOESNOTSTART**

| The predicate evaluates true when the resource that is specified by the
| command does not start with the value of the PRIMPRED option.

| **EQUALS**

| The predicate evaluates true when the resource that is specified by the
| command is equal to the value of the PRIMPRED option.

| **GREATERTHAN**

| The predicate evaluates true when the resource that is specified by the
| command is greater than the value of the PRIMPRED option.

| **ISNOTGREATER**

| The predicate evaluates true when the resource that is specified by the
| command is equal to or less than the value of the PRIMPRED option.

| **ISNOTLESS**

| The predicate evaluates true when the resource specified by the
| command is equal to or greater than the value of the PRIMPRED option.

| **LESSTHAN**

| The predicate evaluates true when the resource that is specified by the
| command is less than the value of the PRIMPRED option.

| **STARTSWITH**

| The predicate evaluates true when the resource that is specified by the
| command starts with the value of the PRIMPRED option.

| **PRIMPREDTYPE** (*cvda*)

| Returns a CVDA value that identifies the type of the primary predicate for this
| capture specification. Possible CVDA values are as follows:

| **CONTAINER**

| The primary predicate is a container.

| **CURRENTPGM**

| The primary predicate is the current program name.

| **EVENT**

| The primary predicate is a CICS event.

| **FILE** The primary predicate is a CICS file.

| **MAP** The primary predicate is a CICS basic mapping support (BMS) map.

| **NONE**

| The capture specification has no primary predicate.

- | **PROGRAM**
| The primary predicate is a CICS program name.
- | **SERVICE**
| The primary predicate is a CICS service or a WEBSERVICE resource.
- | **TDQUEUE**
| The primary predicate is a CICS transient data queue.
- | **TRANCLASS**
| The primary predicate is a CICS transaction class name.
- | **TRANSACTION**
| The primary predicate is a CICS transaction identifier.
- | **TSQUEUE**
| The primary predicate is a CICS temporary storage queue.

Conditions

END

RESP2 values:

- 2 There are no more resource definitions of this type.
- 8 The event binding has been deleted so browse has been terminated early.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

INVREQ

- 4 An EVENTBINDING name has not been specified for the START CAPTURESPEC browse.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to read the event binding.

NOTFND

RESP2 values:

- 2 The specified capture specification cannot be found.
- 3 The specified event binding cannot be found.

INQUIRE CFDTPOOL

Retrieve information about a coupling facility data table pool.

INQUIRE CFDTPOOL

►►—INQUIRE CFDTPOOL(*data-value*)—┐
 └──CONNSTATUS(*cvda*)──┘

Conditions: END, ILLOGIC, NOTAUTH, POOLERR

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The INQUIRE CFDTPOOL command returns the status of the connection of the local CICS region to a coupling facility data table pool.

Browsing

You can also browse through all the coupling facility data table pool names that a CICS region has installed, using the browse options (START, NEXT, and END) on INQUIRE CFDTPOOL commands. CICS implicitly installs coupling facility data table pool names from file definitions that specify a coupling facility data table pool name, even if the definition specifies TABLE(NO). See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

Options

CFDTPOOL (*data-value*)

specifies the 8-character name of the coupling facility data table pool about which you are inquiring.

CONNSTATUS (*cvda*)

returns a CVDA value indicating whether CICS is connected to the specified pool.

CVDA values are:

CONNECTED

The server for the coupling facility data table pool is available in this MVS image, and this CICS is currently connected to it.

UNCONNECTED

The server for the coupling facility data table pool is available in this MVS image, but this CICS is not currently connected to it.

UNAVAILABLE

The server for the coupling facility data table pool is currently unavailable in this MVS image.

Conditions

END

RESP2 values:

2 There are no more coupling facility data table pools to browse.

ILLOGIC

RESP2 values:

1 You have issued a START command when a browse of CFDTPOOLS is already in progress, or you have issued a NEXT or an END command when a browse of CFDTPOOLS is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the task issuing the command is not authorized to use this command.

POOLERR

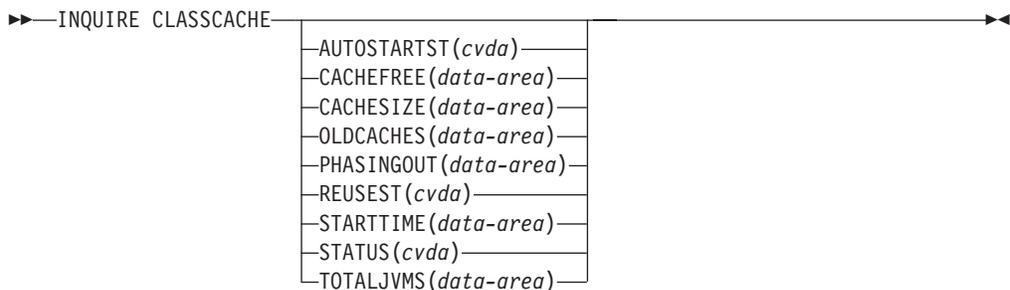
RESP2 values:

- 1 The named CFDT pool was not found. Either CICS has not installed any file definitions that specify the named coupling facility data table pool, or the name is specified incorrectly on the command.
- 2 An internal control structure that CICS uses to maintain access to CFDT pools has been altered while the set of pools known to CICS was being browsed.

INQUIRE CLASSCACHE

Retrieve information about the shared class cache for pooled JVMs in the CICS region, and report the presence of any old shared class caches that are awaiting deletion.

INQUIRE CLASSCACHE



Conditions: NOTAUTH

This command is threadsafe.

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The **INQUIRE CLASSCACHE** command returns information about the shared class cache for pooled JVMs in the CICS region. A CICS region has only one active shared class cache for pooled JVMs; therefore, no name or identifier is required on this command. This command does not return information about class caches that are being used by JVM servers.

Options

AUTOSTARTST (cvda)

Returns a CVDA value indicating the status of autostart for the shared class cache. CVDA values are as follows:

ENABLED

When autostart is enabled, if the shared class cache has been stopped or has not yet been started on this CICS run, the shared class cache is

started as soon as CICS receives a request to run a Java application in a JVM with a profile that requires the use of the shared class cache.

DISABLED

When autostart is disabled, if the shared class cache has been stopped or has not yet been started on this CICS run, an explicit **PERFORM CLASSCACHE** command is required to start it. If the status of the shared class cache is STOPPED and autostart is disabled, and CICS receives a request to run a Java application in a JVM with a profile that requires the use of the shared class cache, the application cannot run.

CACHEFREE(*data-area*)

Returns a doubleword binary value representing the amount of free space in the shared class cache, in bytes. CACHEFREE returns zero until the first JVM has run in the shared class cache.

CACHESIZE(*data-area*)

Returns a doubleword binary value representing the size of the shared class cache, in bytes.

- If the status of the shared class cache is STOPPED, this value is the default size that is used when the shared class cache is started.
- If the status of the shared class cache is STARTING or STARTED, this value is the size of the current shared class cache.
- If the status of the shared class cache is RELOADING, this value is the size of the new shared class cache that is being loaded.

OLDCACHES(*data-area*)

Returns a fullword binary value giving the number of old shared class caches that are still present in the region because they are waiting for JVMs that are using them to be phased out. If the status of the current shared class cache is STOPPED, and JVMs are still using it, that shared class cache is included in the number of old shared class caches.

PHASINGOUT(*data-area*)

Returns a fullword binary value giving the number of JVMs that are using an old shared class cache and are being phased out. If the status of the current shared class cache is STOPPED, any JVMs that are still using it are included in the number of JVMs being phased out.

REUSEST(*cvda*)

Returns a CVDA value indicating the reuse status of the pooled JVMs that use the shared class cache.

REUSE

The JVMs that use the shared class cache are continuous. In this CICS release, JVMs associated with the shared class cache are always continuous (reusable) JVMs.

UNKNOWN

The shared class cache is not started. Although CICS cannot identify the reuse status in this situation, the status always becomes REUSE when the shared class cache is started.

STARTTIME(*data-area*)

Returns the time when the current shared class cache was started. The time is in CICS ABSTIME format. You can convert this absolute date and time to an alternative format by using the **FORMATTIME** command.

STATUS(*cvda*)

Returns the status of the current shared class cache. CVDA values are as follows:

STARTING

The shared class cache is being initialized. New JVMs that require the use of the shared class cache must wait until the startup process is complete and the shared class cache is ready. If initialization of the shared class cache is unsuccessful, any waiting requests for JVMs that require the use of the shared class cache fail.

STARTED

The shared class cache is ready, and it can be used by JVMs.

RELOADING

A new shared class cache is being loaded to replace the existing shared class cache. JVMs that are using the shared class cache, both those that were already allocated to tasks and those that were allocated to tasks after the command was issued, continue to use the existing shared class cache until the new shared class cache is ready.

STOPPED

The shared class cache has either not been initialized on this CICS run or it has been stopped by a **PERFORM CLASSCACHE** command or a **SET JVMPPOOL** command. If autostart is disabled, requests to run a Java application in a pooled JVM with a profile that requires the use of the shared class cache fail. If autostart is enabled, a new shared class cache is initialized as soon as CICS receives a request to run a Java application in a pooled JVM with a profile that requires the use of the shared class cache.

TOTALJVMs(*data-area*)

Returns a fullword binary value giving the number of JVMs in the CICS region that are using a shared class cache. This value includes both the pooled JVMs that are using the current shared class cache and any pooled JVMs that are using an old shared class cache and are being phased out.

Conditions**NOTAUTH**

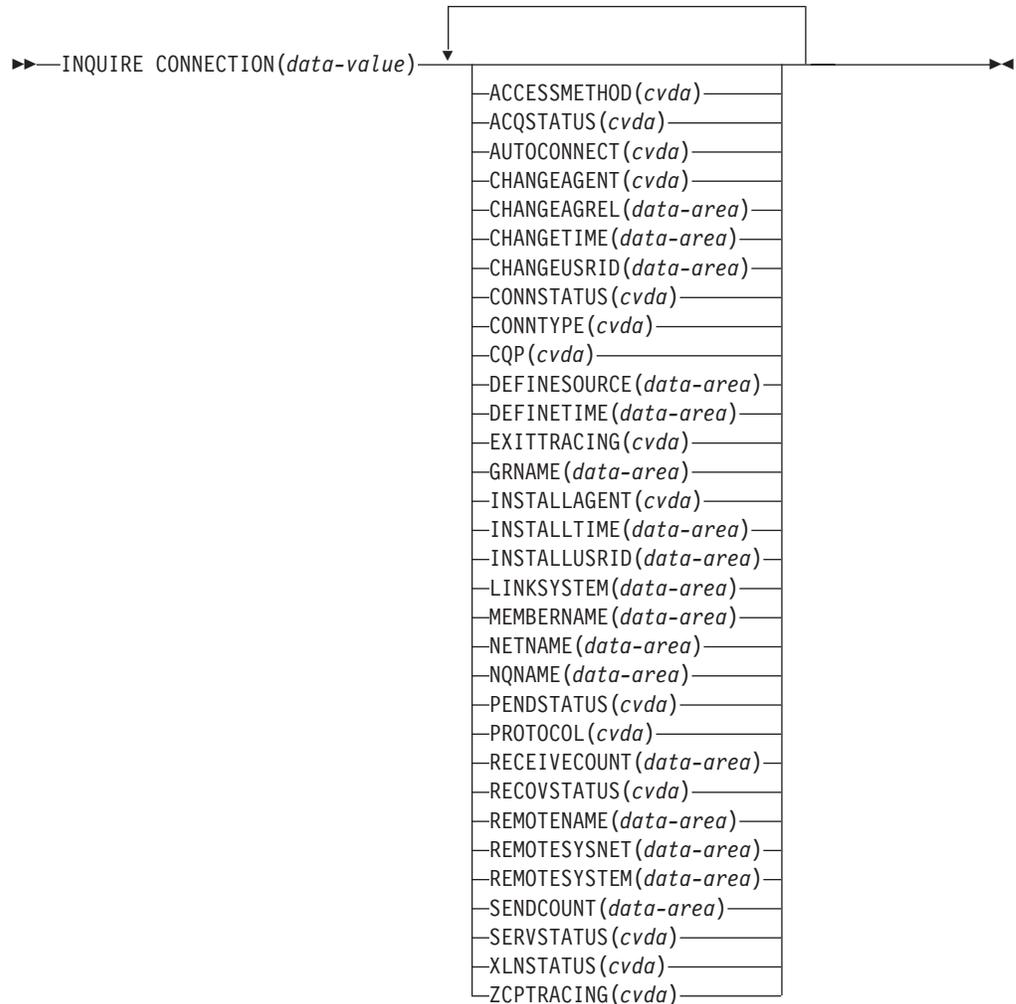
RESP2 value:

100 The user associated with the issuing task is not authorized to use this command.

INQUIRE CONNECTION

Retrieve information about an MRO or ISC over SNA connection to a remote system.

INQUIRE CONNECTION



Conditions: END, ILLOGIC, NOTAUTH, SYSIDERR

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The INQUIRE CONNECTION command retrieves information about a connection from your local CICS region to another CICS region or another system.

Note: INQUIRE CONNECTION returns information about MRO and ISC over SNA connections. See also “INQUIRE IPCONN” on page 348. The INQUIRE IPCONN command returns information about IPIC connections (also known as IPCONNs).

For information about the different kinds of intercommunication connections, see the *CICS Intercommunication Guide*.

Browsing

You can also browse through all the CONNECTION definitions installed in your system by using the browse options, START, NEXT, and END, on INQUIRE

CONNECTION commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

ACCESSMETHOD(*cvda*)

Returns a CVDA value indicating the type of connection between the local system and the one about which you are inquiring. CVDA values are as follows:

INDIRECT

Communication between the local CICS system and the system defined by this connection is through the system named in the INDSYS operand of the CONNECTION definition.

IRC The connection is used for multiregion operation (MRO) and has been defined to use DFHIRP for communication. If the CONNSTATUS is ACQUIRED, the MRO partner is running on the same MVS image. If the CONNSTATUS is RELEASED, the MRO partner might not be on the same MVS image; if it is not, the XCF access method is used when the connection becomes ACQUIRED.

VTAM (now z/OS Communications Server)

The connection is used for intersystem communication (ISC).

XCF The connection is used for multiregion operation (MRO), and communication uses the cross-system coupling facility (XCF) of z/OS. XCF is used for MRO links between CICS regions on different MVS images in a z/OS sysplex. It is selected dynamically by CICS for such links when the access method is defined as IRC or XM in the CONNECTION definition.

XM The connection is used for multiregion operation (MRO) and has been defined to use MVS cross-memory (XM) services for communication. If the CONNSTATUS is ACQUIRED, the MRO partner is running on the same MVS image. If the CONNSTATUS is RELEASED, the MRO partner might not be on the same MVS image; if it is not, the XCF access method is used when the connection becomes ACQUIRED.

ACQSTATUS(*cvda*) (APPC only)

Returns the same value as the CONNSTATUS option and is retained only for compatibility purposes. Use CONNSTATUS in new applications.

AUTOCONNECT(*cvda*) (z/OS Communications Server only)

Returns a CVDA value identifying which AUTOCONNECT option has been specified in the CONNECTION definition. For parallel APPC connections (those with SINGLESESS(NO) specified), the AUTOCONNECT operand

controls the binding of the LU services manager sessions whenever communication with z/OS Communications Server is started. For single-session APPC connections and for LUTYPE6.1 connections, the AUTOCONNECT operand on the CONNECTION definition is ignored and the value returned is not meaningful. CVDA values are as follows:

ALLCONN

AUTOCONNECT(ALL) has been specified on the CONNECTION definition. This specification is the same as specifying AUTOCONNECT(YES), but it can be used for consistency with the associated SESSIONS definition, which allows AUTOCONNECT(ALL).

AUTOCONN

AUTOCONNECT(YES) has been specified on the CONNECTION definition. CICS will try to bind the LU services manager sessions.

NONAUTOCONN

AUTOCONNECT(NO) has been specified for the CONNECTION definition. CICS does not bind LU services manager sessions.

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

AUTOINSTALL

The resource was autoinstalled.

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see FORMATTIME in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

CONNECTION (*data-value*)

Specifies the 4-character identifier of the remote system or region about which you are inquiring; that is, the name assigned to its CONNECTION definition.

CONNSTATUS (*cvda*) (**APPC and MRO only**)

Returns a CVDA value identifying the state of the connection between CICS and the remote system. The remote system can be an APPC partner or a CICS MRO partner; CONNSTATUS is not applicable to EXCI or LU6.1 connections.

The ACQUIRED and RELEASED CVDA values are common to both APPC and MRO; the others are unique to APPC. CVDA values are as follows:

ACQUIRED

The connection is acquired. These criteria apply to ACQUIRED for z/OS Communications Server links:

- The partner LU has been contacted.
- The initial CHANGE-NUMBER-OF-SESSIONS (CNOS) exchange has been done.

These criteria apply to ACQUIRED for MRO links:

- Both sides of the link are in service.
- Both sides of the link are successfully logged on to DFHIRP.
- A connection request by each side has been successful for at least one session, and therefore each side can send and receive data.

AVAILABLE (APPC only)

The connection is acquired but no sessions are currently bound because they were unbound for limited resource reasons.

FREEING (APPC only)

The connection is being released.

NOTAPPLIC

The connection is not a CICS-to-CICS MRO connection or an APPC connection.

OBTAINING (APPC only)

The connection is being acquired. The connection remains in the OBTAINING state until all the criteria for ACQUIRED have been met.

RELEASED

The connection is RELEASED. Although it might also be in INSERVICE status, it is not usable.

The RELEASED status can be caused by any one of a number of general conditions:

- The remote system has not yet initialized.
- No CONNECTION definition exists on the remote system.
- The connection on the remote system has been set out of service.

In the case of a CICS-to-CICS MRO connection, the RELEASED status might also be because of these reasons:

- The remote CICS region has not yet logged on to DFHIRP.
- The remote CICS region has closed interregion communication.

In the case of an APPC ISC connection, the RELEASED status might also be because of these reasons:

- The remote CICS region has not yet opened its z/OS Communications Server ACB.
- AUTOCONNECT(NO) has been specified on the CONNECTION or SESSIONS definition.

CONNTYPE(*cvda*) (EXCI only)

Returns a CVDA value identifying the type of external CICS interface (EXCI) sessions, or pipes, defined for this connection. This option applies only to EXCI connections. CVDA values are as follows:

GENERIC

The connection is generic. A GENERIC connection is an MRO link with many sessions to be shared by multiple users.

NOTAPPLIC

The connection is not an EXCI connection.

SPECIFIC

The connection is specific. A SPECIFIC connection is an MRO link with one or more sessions dedicated to a single user.

See the *CICS External Interfaces Guide* for more information about EXCI connections.

CQP(*cvda*)

Returns a CVDA indicating the status of the connection quiesce protocol for the connection. The CVDA values are as follows:

COMPLETE

The quiesce protocol completed successfully when the connection was released. This value reverts to UNATTEMPTED if the connection is reacquired.

FAILED

The protocol failed for one of several reasons, such as a session failure during execution of the protocol or because the partner receiving the CQP flow has outstanding work.

UNATTEMPTED

The connection supports the protocol, but it has not yet been invoked because the connection status is ACQUIRED.

NOTSUPPORTED

The connection does not support the quiesce protocol because, for example, the partner is a back-level CICS region that does not support the connection quiesce protocol.

DEFINESOURCE(*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

EXITTRACING(*cvda*) (**z/OS Communications Server only**)

Returns a CVDA value indicating whether the terminal exit program is tracing the sessions associated with this connection. CVDA values are as follows:

EXITTRACE

Tracing is on.

NOEXITTRACE

Tracing is off.

NOTAPPLIC

The connection is not LU6.1 or APPC.

GRNAME(*data-area*)

Returns, for an APPC connection to a generic resource when this system is also a generic resource, the 8-character generic resource name of the connected LU.

Otherwise, it returns blanks. CICS assumes that the partner is a generic resource if the two NETNAMEs sent with a BIND are different. This information can also be returned for a partner that is not a generic resource but uses XRF.

INSTALLAGENT(*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

AUTOINSTALL

The resource was autoinstalled.

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

LINKSYSTEM(*data-area*)

Returns the 4-character name of the connection that is the real link towards the TOR for a remote or indirect system entry, if it is available. It is not set if some connection definitions in the chain from the remote or indirect entry to the link system are missing.

MEMBERNAME(*data-area*)

Returns, for an APPC connection to a generic resource when this system is also a generic resource, the 8-character member name (APPLID) of the connected LU. Otherwise, it returns blanks. CICS assumes that the partner is a generic resource if the two NETNAMEs sent with a BIND are different. This information can also be returned for a partner that is not a generic resource but uses XRF.

NETNAME(*data-area*)

Returns, from the NETNAME value specified in the CONNECTION definition, the 8-character name by which the remote system is known to the network.

For an ISC connection, the NETNAME corresponds to the z/OS Communications Server APPLID of the remote system.

For a CICS-to-CICS MRO connection, the NETNAME is the name that the remote system uses to log on to DFHIRP (from the system initialization APPLID option).

For a SPECIFIC EXCI connection, NETNAME is the name of the client program that is passed on the EXCI INITIALIZE_USER command; for a GENERIC EXCI connection, NETNAME is always blanks.

For an indirect connection, NETNAME corresponds to the APPLID, as specified in the system initialization APPLID option, of the terminal-owning region.

NQNAME (*data-area*)

Returns the 17-character network-qualified name for any connection that received an NQNAME from z/OS Communications Server at logon time.

NQNAME, which is supported for problem determination purposes only, is returned for both autoinstalled and RDO-defined resources if it has been supplied by z/OS Communications Server. However, it is not catalogued for RDO-defined resources and is therefore not available on a restart until that resource logs on again.

If the resource is not z/OS Communications Server, NQNAME is blank. If the resource is a z/OS Communications Server resource but has not yet received an NQNAME, CICS returns the known netname.

PENDSTATUS (*cvda*) (**APPC and MRO only**)

Returns a CVDA value identifying whether there are any pending units of work for this connection. CVDA values are as follows:

NOTAPPLIC

This session is not an APPC parallel-session or a CICS-to-CICS MRO connection.

NOTPENDING

No mismatch of lognames with the partner has occurred.

Note: MRO connections to systems earlier than CICS Transaction Server for OS/390® do not use lognames. Therefore, for these connections, PENDSTATUS always returns NOTPENDING.

PENDING

Resynchronization work is outstanding for the connection, but the partner system has performed an initial start, preventing completion of the resynchronization process. (If the partner system is a CICS version earlier than CICS Transaction Server for OS/390, a cold start performed on the partner system has the same effect.) You can use the SET CONNECTION NOTPENDING command to unilaterally commit or back out the units of work associated with the connection, according to their associated transaction definitions. You can also investigate the units of work individually and force them to commit or back out, in which case you must also complete the recovery activity by using a SET CONNECTION NOTPENDING command to clear the PENDING condition.

If this connection is an APPC connection, no new sync point work (that is, work involving sync level 2 protocols) can be transmitted across it until a SET CONNECTION NOTPENDING command has been issued. This restriction does not apply to MRO connections.

If you are not concerned by the loss of synchronization caused by the initial or cold start of the partner, you can cause the SET CONNECTION NOTPENDING command to be issued automatically by specifying XLNACTION(FORCE) on the CONNECTION definition.

For further information about pending units of work, see the *CICS Intercommunication Guide*.

PROTOCOL (*cvda*) (**z/OS Communications Server and EXCI only**)

Returns a CVDA value identifying the protocol in use if this is a z/OS Communications Server or EXCI connection. CVDA values are as follows:

APPC The connection uses the z/OS Communications Server LUTYPE6.2 protocol for intersystem communication.

EXCI The connection uses the external CICS interface for communication between CICS and a non-CICS client program.

LU61 The connection uses the z/OS Communications Server LUTYPE6.1 protocol.

NOTAPPLIC

The connection is used for CICS-to-CICS MRO communication or it is INDIRECT.

RECEIVECOUNT(*data-area*) (MRO only)

Returns a fullword binary value giving the number of RECEIVE sessions defined for this connection. This option applies only to MRO connections; for others, the value returned is -1.

RECOVSTATUS(*cvda*) (APPC and MRO only)

Returns a CVDA value indicating whether resynchronization work is outstanding for the connection. The connection might never have been connected, have been quiesced and all resynchronization work completed, or disrupted without quiesce, in which case resynchronization might be necessary. CVDA values are as follows:

NORECOVDATA

Neither side has recovery information outstanding.

NOTAPPLIC

This session is not an APPC parallel-session or a CICS-to-CICS MRO connection, and it does not support 2-phase commit protocols.

NRS CICS does not have recovery outstanding for the connection, but the partner might have.

RECOVDATA

Indoubt units of work are associated with the connection, or outstanding resyns are awaiting FORGET on the connection. Resynchronization takes place when the connection next becomes active or when the UOW is unshunted.

If recovery is outstanding, on completion of exchange lognames either resynchronization takes place or, in the case of a cold exchange, the PENDING condition is created.

REMOTENAME(*data-area*)

Returns the 4-character name by which this connection is known in a remote system, if the subject of the inquiry is a remote connection.

REMOTESYSNET(*data-area*)

Returns the 8-character netname of the owning TOR, if the subject of this inquiry is a remote connection. If it is blank, but the connection is remote, the system named in the REMOTESYSTEM field has not been installed, and no value was specified for the REMOTESYSNET option when the connection was defined.

REMOTESYSTEM(*data-area*)

Returns the 4-character name of a connection, if the subject of the inquiry is a remote connection. The named connection can be either a connection entry that links towards the TOR or an indirect connection, which provides the netname of the TOR, and itself points to another connection.

Otherwise this field is blank.

SENDCOUNT(*data-area*) (**MRO only**)

Returns a fullword binary value giving the number of SEND sessions defined for this connection. For EXCI connections, the SENDCOUNT is always zero. This option applies only to MRO connections; for others, the value returned is -1.

SERVSTATUS(*cvda*)

Returns a CVDA value indicating whether data can be sent and received on the connection. CVDA values are as follows:

GOINGOUT

OUTSERVICE has been requested on a SET CONNECTION command, and the request cannot be acted on until some current work has completed.

INSERVICE

Data can be sent and received.

OUTSERVICE

Data cannot be sent and received.

XLNSTATUS(*cvda*) (**APPC only**)

Returns a CVDA value identifying the status of the exchange log names (XLN) process. CVDA values are as follows:

NOTAPPLIC

The XLN process is not applicable because the link is in one of these states:

- Is released
- Is MRO, LUTYPE6.1, or single-session APPC
- Does not support synchronization level 2 conversations.

For information about the APPC exchange log names process, see the *CICS Intercommunication Guide*.

XNOTDONE

The XLN flow for the APPC connection has not completed successfully. The CSMT log can contain information relating to this state. Synchronization level 2 conversations are not allowed on the connection, but synchronization levels 0 and 1 are still allowed.

XOK The XLN process for the APPC connection has completed successfully.

ZCPTRACING(*cvda*) (**z/OS Communications Server only**)

Returns a CVDA value indicating whether the z/OS Communications Server control component of CICS is tracing activity on the sessions associated with this connection. CVDA values are as follows:

NOTAPPLIC

The connection is not LUTYPE6.1 or APPC.

NOZCPTRACE

ZCP tracing is not active.

ZCPTRACE

ZCP tracing is active.

Conditions**END**

RESP2 values:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SYSIDERR

RESP2 values:

- 1 The connection cannot be found.

Remote connections

In addition to links to other systems or devices, some connection definitions refer to remote connections. A remote connection is a link to another system or device that is owned by another CICS system.

The owning system in a remote connection is called the terminal-owning region (TOR). Note that different connections might have different TORs; that is, different owning systems for a particular connection.

The way that the local system is connected to the TOR makes a difference to the way in which the REMOTESYSTEM and REMOTESYSNET options of the remote CONNECTION definition are specified.

If the TOR is *directly* connected to the local system, the REMOTESYSTEM field typically names the CONNECTION definition or first four characters of the IPCONN definition for the link.

If a remote connection is on a system that is *not directly linked* to the TOR and you are using an IPCONN resource definition, you must uniquely identify the TOR by specifying the netname of the TOR.

If a remote connection is on a system that is *not directly linked* to the TOR, and you are using a CONNECTION resource definition, the REMOTESYSTEM option can name one of two types of connection:

- A “real” connection that is the next link in the chain towards the TOR. In this case, the REMOTESYSNET option must specify the netname of the TOR. For example, in “Example of remote definitions” on page 269, connection CON2 points to connection INTS, which is the first link in the chain to the TOR.
- An indirect connection. In this case, the indirect connection NETNAME contains the netname of the TOR, and its INDSYS option names another connection, which can also be indirect or “real”. For example, in “Example of remote definitions” on page 269, connection CON1 points to connection INDC, which in turn points to INTS. The REMOTESYSNET option of the remote CONNECTION definition might or might not specify the netname of the TOR.

Note these points about remote connection definitions:

- The LINKSYSTEM option of INQUIRE CONNECTION returns the “real” connection that is the next link towards the TOR. It is determined by looking at the logical chain of entries from the connection in question to the “real”

connection entry. If the chain is broken, because an entry has not been installed yet, or has been discarded, LINKSYSTEM is not set.

- The REMOTESYSNET option of INQUIRE CONNECTION always returns the netname of the TOR, even if the REMOTESYSNET option was not specified on the original CONNECTION definition, unless both the following statements are true:
 - The system returned in the REMOTESYSTEM field has not been installed.
 - REMOTESYSNET was not specified on the CONNECTION or IPCONN definition.

Example of remote definitions

An application-owning region (AOR) has direct links to an intermediate system, CONNECTION(INTS) NETNAME(SYSI) to CONNECTION(AOR) NETNAME(AOR1); and also to terminal-owning regions (TORs), CONNECTION(TORB) NETNAME(TOR2) to CONNECTION(AOR) NETNAME(AOR1).

The AOR, APPLID=AOR1, has remote connection definitions, CONNECTION(CON1) NETNAME(NETCON1) REMOTESYSTEM(INDC) REMOTESYSNET() and links by indirect link to CONNECTION(INDC) NETNAME(TOR1) INDSYS(INTS); and then links by direct links to the intermediate system, CONNECTION(INTS) NETNAME(SYSI) to CONNECTION(AOR) NETNAME(AOR1). The AOR also has remote terminal definition, TERMINAL(TER1) NETNAME(NETTER1) REMOTESYSTEM(TORB) REMOTESYSNET() that has direct links to the TORs, CONNECTION(TORB) NETNAME(TOR2) to CONNECTION(AOR) NETNAME(AOR1).

The intermediate system has direct links to the TORs, CONNECTION(TORA) NETNAME(TOR1) to CONNECTION(INTS) NETNAME(SYSI).

The TORs have local connection definitions, CONNECTION(CON1) NETNAME(NETCON1), and local terminal definition, TERMINAL(TER1) NETNAME(NETTER1).

The following diagram illustrates a graphical representation of the example above.

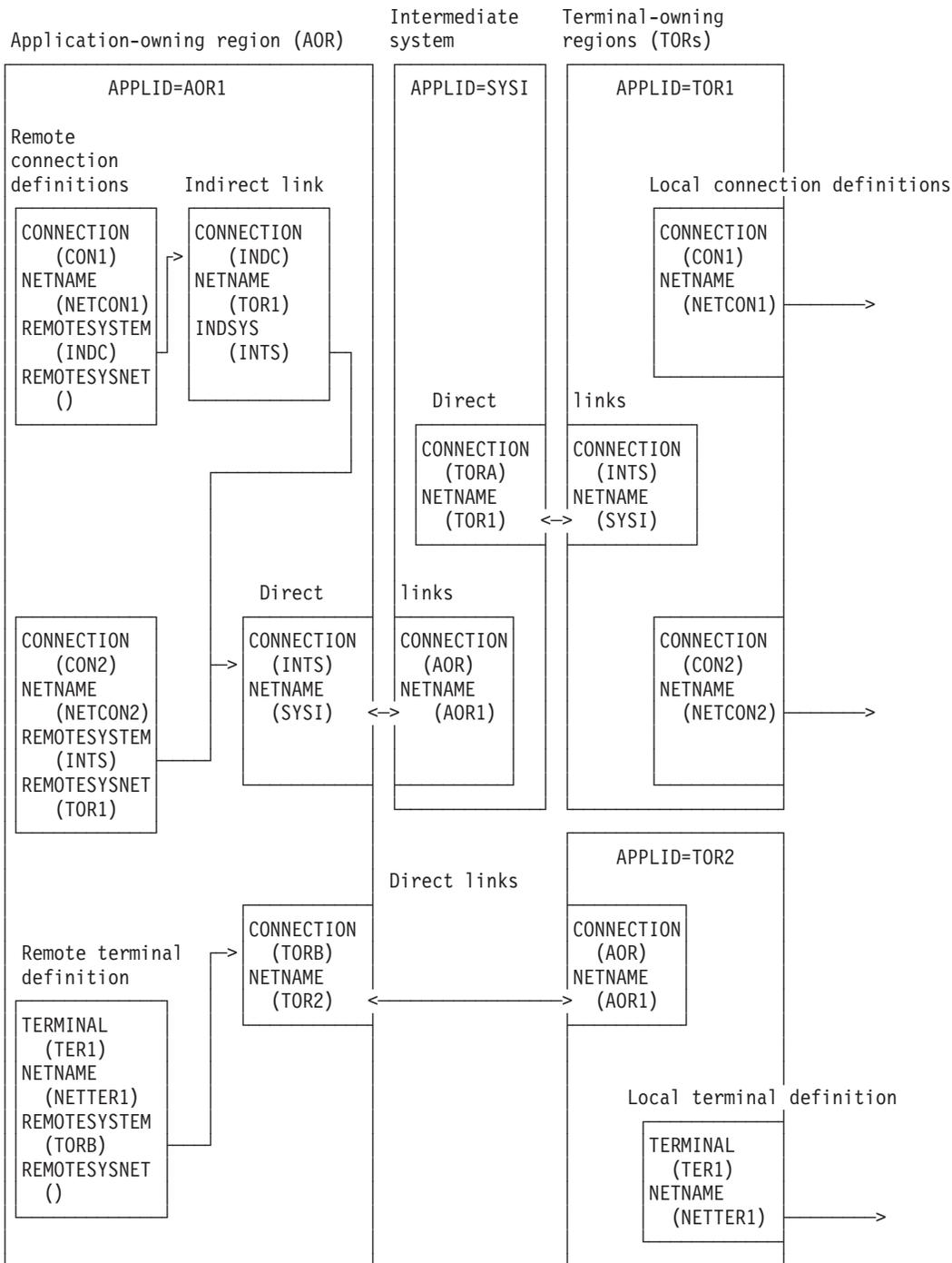
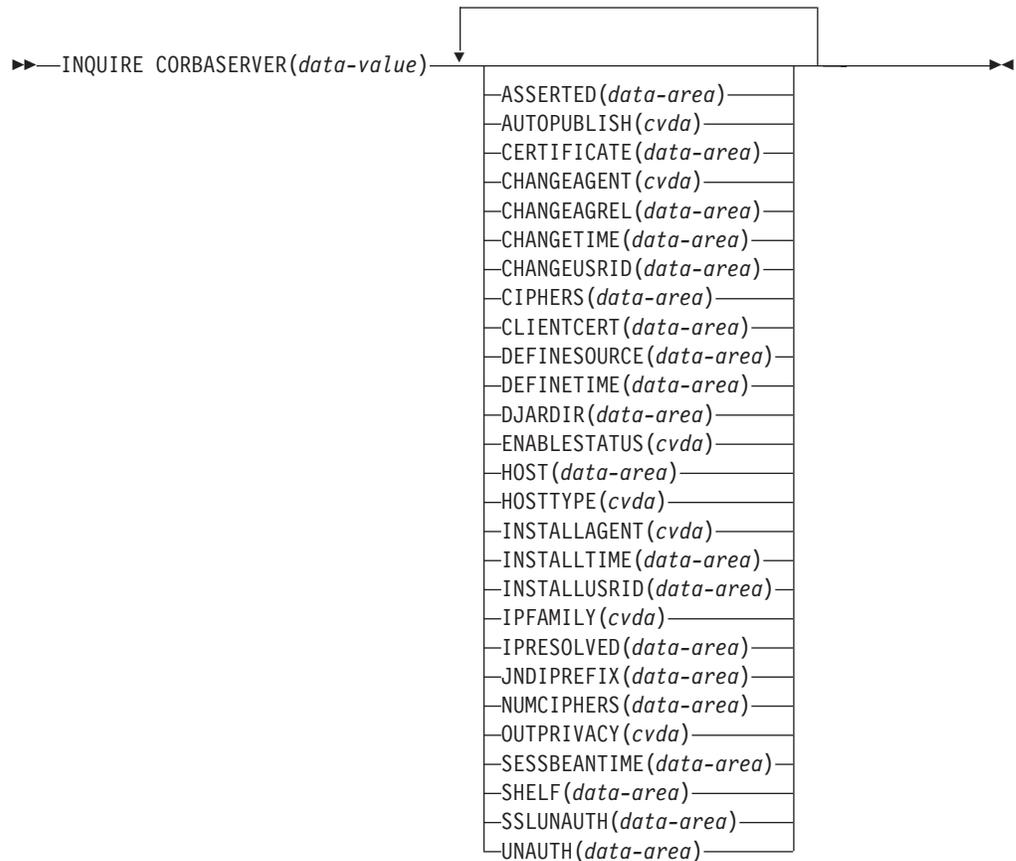


Figure 1. Remote definitions. How the REMOTESYSTEM and REMOTESYSNET options of the CONNECTION definition are specified depends on the path to the TOR. The LINKSYSTEM field of INQUIRE CONNECTION always returns the real link sysid. The REMOTESYSNET field of INQUIRE CONNECTION always returns the netname of the TOR, if it exists.

INQUIRE CORBASERVER

Retrieve information about a particular CorbaServer.

INQUIRE CORBASERVER



Conditions: END, NOTAUTH, NOTFND

For more information about the use of CVDAs, see CICS-value data areas (CVDA) in CICS System Programming Reference.

Description

Use the INQUIRE CORBASERVER command to retrieve information about a particular CorbaServer.

Browsing

You can also browse through all the CorbaServers installed in the region, using the browse options (START, NEXT, and END) on INQUIRE CORBASERVER commands. See for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID,

DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

ASSERTED(*data-area*)

Returns the 8-character name of a TCPIPSERVICE resource that defines the characteristics of the port that is used for inbound IIOP with asserted identity authentication.

AUTOPUBLISH(*cvda*)

Returns a CVDA value indicating whether enterprise beans are to be automatically published to the JNDI namespace when the deployed JAR file that contains them is successfully installed in the CorbaServer. The CVDA values are as follows:

AUTOPUB

Enterprise beans are to be automatically published.

NOAUTO

Enterprise beans are not to be automatically published.

CERTIFICATE(*data-area*)

Returns a 56-character area containing the label of the certificate in the key ring that is used as a client certificate in the SSL handshake for outbound IIOP connections. If the label is blank, the certificate nominated as the default for the key ring is used.

The distinguished name in this certificate is used to provide inputs to the Distinguished Name URM (DFHEJDNX). See the *CICS Customization Guide*.

CHANGEAGENT(*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL(*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see FORMATTIME in CICS Application Programming.

CHANGEUSRID(*data-area*)

Returns the 8-character user ID that ran the change agent.

CIPHERS(*data-area*)

Returns a 56-character area containing the list of cipher suites that is used to negotiate with clients during the SSL handshake. The list is set by the ENCRYPTION system initialization parameter, but you can edit the list to remove or change the order of cipher suites. See the *CICS RACF Security Guide*.

CLIENTCERT(*data-area*)

Returns the 8-character name of a TCPIPSERVICE resource that defines the characteristics of the port that is used for inbound IOP with SSL client certificate authentication.

CORBASERVER(*data-value*)

Specifies the 4-character name of a CorbaServer.

DEFINESOURCE(*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

DJARDIR(*data-value*)

Returns a 255-character area containing the name of the deployed JAR file directory (also known as the *pickup directory*) on HFS. (The pickup directory is the place that you put deployed JAR files that you want to be installed into the CorbaServer by the CICS scanning mechanism.)

ENABLESTATUS(*cvda*)

Returns a CVDA value indicating the current state of the CorbaServer. The CVDA values are as follows:

DISABLED

The CorbaServer is currently not processing any requests and cannot accept new requests. It might have failed to initialize properly or is explicitly disabled.

DISABLING

The CorbaServer is quiescing before entering disabled state. It is not accepting new requests but is allowing current work complete.

DISCARDING

A DISCARD request has been received for this CorbaServer. The CorbaServer is quiescing before being discarded. It is not accepting new requests but is allowing current work to complete.

ENABLED

The CorbaServer is available and is accepting requests.

ENABLING

The CorbaServer is being initialized. It is not yet ready to accept requests.

HOST(*data-area*)

Returns a 255-character area, containing the IP host name or a string containing the dotted decimal or colon hexadecimal IP address, which is included in Interoperable Object References (IORs) exported from the CorbaServer. HOST displays all IPv4 addresses as native IPv4 dotted decimal addresses; for example, 1.2.3.4, regardless of the type of address format used.

You can specify IPv4 and IPv6 addresses in a number of acceptable formats. See IP addresses in the Internet Guide for more information on address formats.

HOST displays the defined host name, IPv4, or IPv6 address.

HOST is specified in the resource definition.

HOSTTYPE(*cvda*)

Returns the address format of the HOST option. HOSTTYPE is set by the domain when the CorbaServer is installed. CVDA values are as follows:

HOSTNAME

The HOST option contains a character host name. The IP address that corresponds to the host name is looked up in the domain name server.

IPV4 The address is specified in IPv4 dotted decimal address format.

IPV6 The address is specified in IPv6 colon hexadecimal address format.

NOTAPPLIC

An incorrect host address was returned (HOST=0.0.0.0).

INSTALLAGENT(*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

IPFAMILY(*cvda*)

Returns the address format of the IPRESOLVED option. CVDA values are as follows:

IPV4 The address is specified in IPv4 dotted decimal address format.

IPV6 The address is specified in IPv6 colon hexadecimal address format.

UNKNOWN

IPRESOLVED is not yet in use or the address cannot be resolved. UNKNOWN is the default when IPRESOLVED is 0.0.0.0

IPRESOLVED(*data-area*)

Returns a 39-character field that specifies the IPv4 or IPv6 address of the HOST option. If the CorbaServer is installed but not enabled, or the address cannot be resolved, a default value of 0.0.0.0 is returned. After the CorbaServer is enabled, IPRESOLVED displays the last resolved IP address that was used by the CorbaServer resource. IPRESOLVED is reset to 0.0.0.0 when the resource is disabled. The content of IPRESOLVED is not recoverable after a warm or emergency restart.

JNDIPREFIX(*data-area*)

Returns a 255-character area containing the JNDI prefix.

NUMCIPHER(*data-area*)

Returns a binary halfword data-area that contains the number of cipher suites that are specified in the CIPHERS attribute.

OUTPRIVACY(*cvda*)

Returns a CVDA value indicating the level of SSL encryption that is used for outbound connections from this CORBASERVER. This level is determined by the CIPHERS attribute. Possible values are as follows:

NOTSUPPORTED

Encryption is not used. During the SSL handshake, CICS advertises only supported cipher suites that do not provide encryption.

REQUIRED

Encryption is used. During the SSL handshake, CICS advertises only supported cipher suites that provide encryption.

SUPPORTED

Encryption is used if both client and server support it. During the SSL handshake, CICS advertises all supported cipher suites.

SESSBEANTIME(*data-area*)

Returns a fullword area containing the elapsed time period in minutes, in the range 0 - 143999 (99 days, 23 hours, 59 minutes) of inactivity after which a session bean can be discarded. A value of zero indicates that beans are not timed out.

SHELF(*data-area*)

Returns a 255-character area containing the name of the HFS shelf directory.

SSLUNAUTH(*data-area*)

Returns the 8-character name of a TCPIP SERVICE resource that defines the characteristics of the port that is used for inbound IIOP with SSL but no client authentication.

UNAUTH(*data-area*)

Returns the 8-character name of a TCPIP SERVICE resource that defines the characteristics of the port that is used for inbound IIOP with no authentication.

Conditions**END**

RESP2 values:

2 There are no more CorbaServers to browse.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

NOTFND

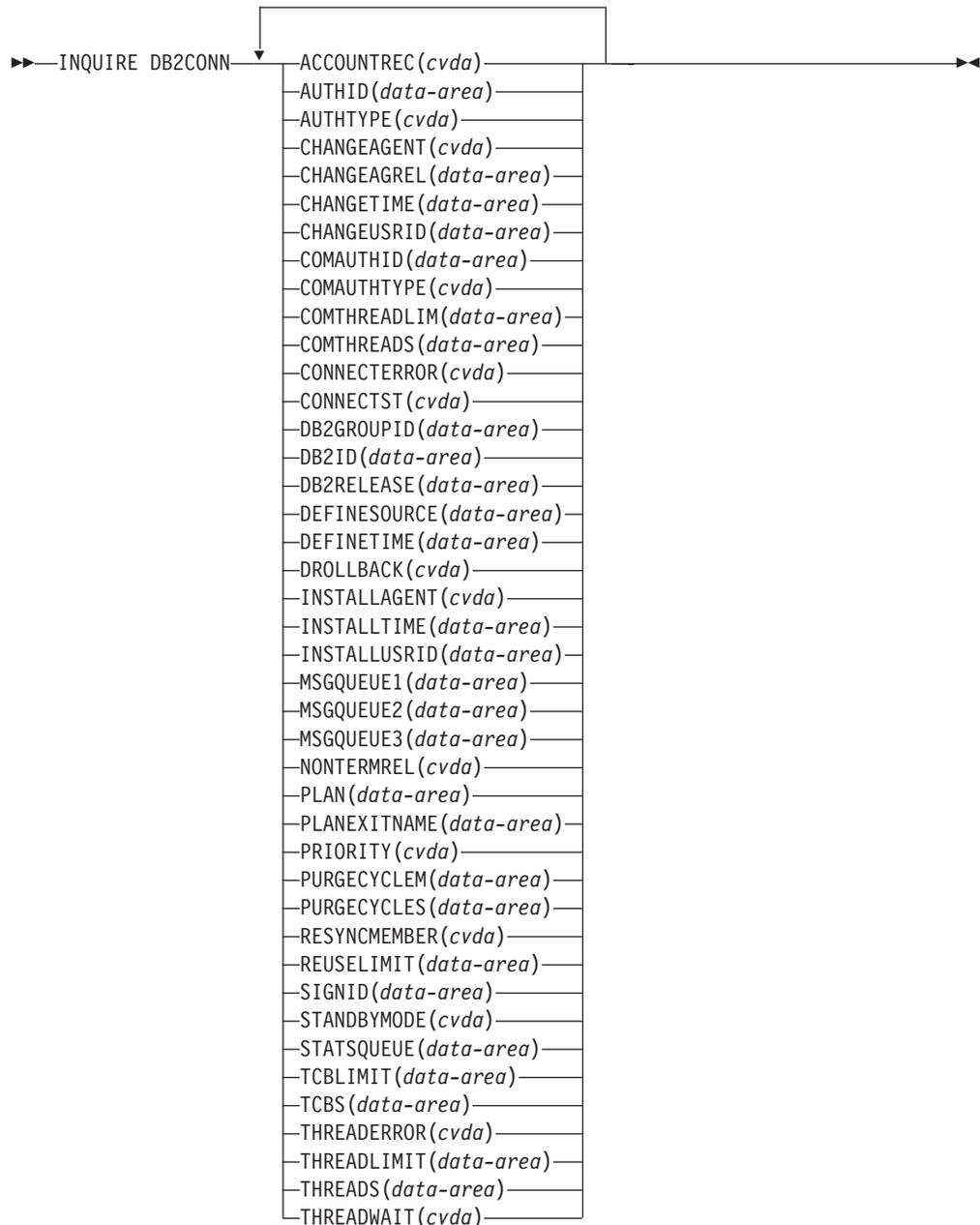
RESP2 values:

2 The named CorbaServer was not found

INQUIRE DB2CONN

Defines the attributes of the connection made between CICS and DB2.

INQUIRE DB2CONN



Conditions: NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the INQUIRE DB2CONN command to inquire about attributes of the currently installed DB2CONN, which defines the connection to DB2.

Because there can be only one DB2CONN installed at a time, the name of the DB2CONN is not required on input.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

ACCOUNTREC (*cvda*)

Returns the minimum amount of DB2 accounting required for transactions using pool threads. The specified minimum can be exceeded as described in the following options. CVDA values are as follows:

UOW The CICS DB2 attachment facility causes an accounting record to be produced by DB2 for each UOW, assuming that the thread is released at the end of the UOW.

TASK The CICS DB2 attachment facility causes a minimum of one accounting record to be produced by DB2 for each CICS task.

A transaction containing multiple UOWs, assuming that the thread is released at sync point, can use a different thread for each of its UOWs. The result might be the production of an accounting record for each UOW. For example, an accounting record is produced if a thread ends after being released or if a thread is reused but the primary AUTHID is changed.

TXID The CICS DB2 attachment facility causes an accounting record to be produced by DB2 when the transaction ID that is using the thread changes.

Because pool threads are typically used by a number of different transaction IDs, there is an increased chance that a transaction containing multiple UOWs uses a different thread for each UOW, assuming that the thread is released at sync point. In this case, an accounting record can be produced for each UOW. For example, an accounting record is produced if a thread ends after being released or if a thread is reused but the primary AUTHID is changed.

NONE

No accounting records are required for transactions using pool threads.

DB2 nevertheless produces at least one accounting record for each thread when the thread is ended. Additionally, authorization changes cause accounting records to be produced.

AUTHID (*data-area*)

Returns an ID to be used for security checking when using pool threads. If an AUTHID is returned, AUTHTYPE does not apply.

AUTHTYPE (*cvda*)

Returns the type of ID to be used for security checking when using pool threads. If an AUTHTYPE is returned, AUTHID is blank. CVDA values are as follows:

GROUP

The 8-character user ID and the connected group name are used as the authorization ID. The following table shows how these two values are interpreted by DB2.

IDs passed to DB2	How DB2 interprets values
CICS sign-on user ID (USERID)	Represents the primary DB2 authorization ID.
RACF-connected group name	If the RACF list of group options is not active, DB2 uses the connected group name supplied by the CICS attachment facility as the secondary DB2 authorization ID. If the RACF list of group options is active, DB2 ignores the connected group name supplied by the CICS attachment facility, but the value appears in the DB2 list of secondary DB2 authorization IDs.

If no RACF group ID is available for this USERID, an 8-character field of blanks is passed to DB2 as the group ID.

SIGN The SIGNID parameter of the DB2CONN is used as the resource authorization ID.

TERM The terminal identification (four characters padded to eight) is used as an authorization ID. An authorization ID cannot be obtained in this manner if a terminal is not connected with the transaction.

If a transaction is started using a CICS command and has no terminal associated with it, do not use AUTHTYPE(TERM).

TX The transaction identification (four characters padded to eight) is used as the authorization ID.

OPID The user operator identification associated with the user ID, associated with the CICS transaction, is used as the authorization ID (three characters padded to eight).

USERID

The 8-character user ID associated with the CICS transaction is used as the authorization ID.

When the DB2 sample sign-on exit DSN3@.SGN is used with AUTHTYPE(USERID), the exit sends the user ID to DB2 as the primary authorization ID and the RACF group ID to DB2 as the secondary ID. When the sample sign-on exit is used, AUTHTYPE(USERID) and AUTHTYPE(GROUP) are the same.

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see FORMATTIME in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

COMAUTHID (*data-area*)

Returns an ID to be used for security checking when using command threads. If COMAUTHID is returned, COMAUTHTYPE is not applicable.

COMAUTHTYPE (*cvda*)

Returns the type of ID to be used for security checking when using command threads. If COMAUTHTYPE is returned, COMAUTHID is blank. CVDA values are as follows:

CGROUP

The 8-character user ID and the connected group name are used as the authorization ID. The following table shows how these two values are interpreted by DB2.

IDs passed to DB2	How DB2 interprets values
CICS sign-on user ID (USERID)	Represents the primary DB2 authorization ID.
RACF-connected group name	If the RACF list of group options is not active, DB2 uses the connected group name supplied by the CICS attachment facility as the secondary DB2 authorization ID. If the RACF list of group options is active, DB2 ignores the connected group name supplied by the CICS attachment facility, but the value appears in the DB2 list of secondary DB2 authorization IDs.

If no RACF group ID is available for this user ID, an 8 character field of blanks is passed to DB2 as the group ID.

CSIGN

The SIGNID parameter of the DB2CONN command is used as the resource authorization ID.

CTERM

The terminal identification (four characters padded to eight) is used as an authorization ID. An authorization ID cannot be obtained in this manner if a terminal is not connected with the transaction.

If a transaction is started using a CICS command and has no terminal associated with it, do not use COMAUTHTYPE(CTERM).

CTX

The transaction identification (four characters padded to eight) is used as the authorization ID.

COPID

The operator identification associated with the user ID that is associated with the CICS transaction is used as the authorization ID (three characters padded to eight).

CUSERID

The 8-character user ID associated with the CICS transaction is used as the authorization ID.

When the DB2 sample sign-on exit DSN3@.SGN is used with COMAUTHTYPE(CUSERID), the exit sends the user ID to DB2 as the primary authorization ID and the RACF group ID to DB2 as the secondary ID. When the sample sign-on exit is used, COMAUTHTYPE(CUSERID) and COMAUTHTYPE(CGROUP) are the same.

COMTHREADLIM(*data-area*)

Returns the current maximum number of command threads that the CICS DB2 attachment allows active before requests overflow to the pool.

COMTHREADS(*data-area*)

Returns the current number of active command threads.

CONNECTERROR(*cvda*)

If CICS is not connected to DB2 because the adapter is in standby mode, describes how this is reported back to an application that has issued a SQL request. CVDA values are as follows:

ABEND

The application is stopped with abend AEY9.

SQLCODE

The application receives a -923 SQLCODE.

CONNECTST(*cvda*)

Returns the status of the CICS DB2 connection. CVDA values are as follows:

CONNECTED

CICS is connected to DB2.

NOTCONNECTED

CICS is not connected to DB2.

CONNECTING

CICS is currently attempting to connect to DB2.

DISCONNING

CICS is currently disconnecting from DB2.

DB2GROUPID(*data-area*)

If you are using group attach, this option returns the name of a data sharing group, or subgroup, of DB2 subsystems that you have specified. CICS attempts to connect to any active member of this group. If the CICS DB2 attachment is connected, the name of the DB2 subsystem that was chosen from the group appears in the DB2ID field. If CICS is waiting to reconnect to a specific DB2 subsystem in the data sharing group, because it is holding outstanding units of work for that subsystem, the name of the specific DB2 subsystem appears in the DB2ID field, and the status CONNECTING is returned. For this situation to arise, RESYNCMEMBER(RESYNC) must be specified.

DB2ID(*data-area*)

If you are not using group attach, this option returns the name of the DB2 subsystem that the CICS DB2 attachment is connected to, or if the CICS DB2

attachment is not connected, the name of the DB2 subsystem that you have specified for CICS to connect to. If you are using group attach and the CICS DB2 attachment is connected, this option returns the name of the DB2 subsystem that the CICS DB2 attachment is connected to. If you are using group attach and the CICS DB2 attachment is not connected, this field is normally blank. However, if CICS is waiting to reconnect to a specific DB2 subsystem, because RESYNCMEMBER(YES) is specified and the DB2 subsystem for which CICS is holding outstanding units of work is unavailable, the command returns the DB2ID value of that subsystem with the status CONNECTING.

DB2RELEASE (*data-area*)

Returns a 4-character value indicating the version and release level of the DB2 subsystem to which CICS is connected. When CICS is not connected to DB2, blanks are returned.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

DROLLBACK (*cvda*)

Returns a value showing whether the CICS DB2 attachment is to initiate a SYNCPOINT ROLLBACK command if a transaction is selected as victim of a deadlock resolution. CVDA values are as follows:

ROLLBACK

The attachment facility issues a sync point rollback before returning control to the application. An SQL return code of -911 is returned to the program.

NOROLLBACK

The attachment facility is not to initiate a rollback for a transaction. An SQL return code of -913 is returned to the application.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID (*data-area*)

Returns the 8-character user ID that installed the resource.

MSGQUEUE1(*data-area*)

Returns the name of the first transient data destination to which unsolicited messages from the CICS DB2 attachment are sent.

MSGQUEUE2(*data-area*)

Returns the name of the second transient data destination to which unsolicited messages from the CICS DB2 attachment are sent.

MSGQUEUE3(*data-area*)

Returns the name of the third transient data destination to which unsolicited messages from the CICS DB2 attachment are sent.

NONTERMREL(*cvda*)

Returns a value showing whether non-terminal transactions are to release threads for reuse at intermediate sync points. CVDA values are as follows:

RELEASE

Non-terminal transactions release threads for reuse at intermediate sync points.

NORELEASE

Non-terminal transactions do not release threads for reuse at intermediate sync points.

PLAN(*data-area*)

Returns the name of the plan used for the pool. If a plan name is returned, PLANEXITNAME is blank.

PLANEXITNAME(*data-area*)

Returns the name of the dynamic plan exit used for pool threads. If a PLANEXITNAME is returned, PLAN is blank.

PRIORITY(*cvda*)

Returns the priority of the pool thread TCBs relative to the CICS main TCB (QR TCB). The thread TCBs are CICS open L8 TCBs. CVDA values are as follows:

HIGH Thread TCBs have a higher priority than the CICS QR TCB.

EQUAL

Thread TCBs have equal priority with the CICS QR TCB.

LOW Thread TCBs have a lower priority than the CICS QR TCB.

PURGECYCLEM(*data-area*)

Returns the number of minutes in the protected thread purge cycle time (the number of seconds is returned by PURGECYCLES). The range for PURGECYCLEM is 0 - 59.

A protected thread is not terminated immediately when it is released. It is terminated only after two completed purge cycles, if it has not been reused in the meantime. For example, if the protected thread purge cycle is set to 30 seconds, a protected thread is purged 30 - 60 seconds after that thread is released. An unprotected thread is terminated when it is released (at sync point or end of task) if there are no other transactions waiting for a thread on that DB2ENTRY.

PURGECYCLES(*data-area*)

Returns the number of seconds in the protected thread purge cycle time (the number of minutes is returned by PURGECYCLEM). The range for PURGECYCLES is 0 - 59. If PURGECYCLEM is zero, the minimum value of PURGECYCLES is 5 seconds.

A protected thread is not terminated immediately when it is released. It is terminated only after two completed purge cycles, if it has not been reused in the meantime. For example, if the protected thread purge cycle is set to 30 seconds, a protected thread is purged 30 - 60 seconds after that thread is released. An unprotected thread is terminated when it is released (at sync point or end of task) if there are no other transactions waiting for a thread on that DB2ENTRY.

RESYNCMEMBER(*cvda*)

This option applies only if you are using group attach, and specifies the strategy that CICS adopts if outstanding units of work are being held for the last DB2 data sharing group member to which CICS was connected. (Units of work that are shunted indoubt are not included in this process, because CICS itself cannot resolve those units of work at this time. Resynchronization for those UOWs occurs when CICS has resynchronized with its remote coordinator.) CVDA values are as follows:

RESYNC

CICS connects to the same DB2 data sharing group member.

NORESINC

CICS makes one attempt to connect to the same DB2 data sharing group member, and, if that attempt fails, CICS connects to any member of the DB2 data sharing group and issues a warning about the outstanding units of work.

NOTAPPLIC

A value of NOTAPPLIC is returned if you are not using group attach.

REUSELIMIT(*data-area*)

Returns a value in the range 0 - 10000 representing the maximum number of times a thread can be reused before it is terminated. The default is 1000. A value of 0 means that there is no limit on the number of times that a thread can be reused. Long-running CICS DB2 threads that are constantly being reused build up resources in DB2 that can cause storage problems.

The reuse limit applies to unprotected threads both in the pool and on a DB2ENTRY, and to protected DB2ENTRY threads.

SIGNID(*data-area*)

Returns the authorization ID to be used by the CICS DB2 attachment when signing on to DB2 for pool and DB2 entry threads specifying AUTHTYPE(SIGN) and command threads specifying COMAUTHTYPE(CSIGN).

STANDBYMODE(*cvda*)

Returns the action to be taken by the CICS DB2 attachment if DB2 is not active when an attempt is made to start the connection from CICS to DB2. CVDA values are as follows:

NOCONNECT

The CICS DB2 attachment ends.

CONNECT

The CICS DB2 attachment goes into standby mode to wait for DB2.

RECONNECT

The CICS DB2 attachment goes into standby mode and waits for DB2. After connecting to DB2, if DB2 later fails, the CICS DB2 attachment reverts to standby mode again and then reconnects to DB2 when it restarts.

STATSQUEUE (*data-area*)

Returns the transient data destination for CICS DB2 attachment statistics produced when the CICS DB2 attachment is shut down.

TCBLIMIT (*data-area*)

Returns the maximum number of TCBs that can be used to process DB2 requests. When connected to DB2 Version 5 or earlier, the CICS DB2 attachment facility creates the TCBs in the form of subtasks up to the limit specified by TCBLIMIT. Each of these subtasks identifies to DB2 and creates a connection into DB2. When connected to DB2 Version 6 or later, CICS creates open TCBs (up to the limit specified in the system initialization parameter MAXOPENTCBS). The TCBLIMIT attribute of the DB2CONN definition governs how many of the open TCBs can be used to access DB2; that is, how many of them can identify to DB2 and create a connection into DB2.

TCBS (*data-area*)

Returns a number indicating the TCBs currently used by the CICS DB2 attachment facility. The number returned is the number of TCBs that are associated with DB2 connections (command, pool, or DB2ENTRY threads), so the interpretation of the number depends on the release of DB2 to which CICS is connected, as follows:

Connected to DB2 Version 5 or earlier (therefore not using the open transaction environment)

Subtask TCBs are created and managed by the CICS DB2 attachment facility to service DB2 requests, and remain permanently associated with DB2 connections (command, pool, or DB2ENTRY threads). In this case, the TCBS option returns the highwater mark of TCBs created to access DB2.

Connected to DB2 Version 6 or later (therefore using the open transaction environment)

The TCBs used by the CICS DB2 attachment facility are allocated by CICS from the pool of L8 mode TCBs. A DB2 connection is not permanently assigned to the same L8 TCB, and, between CICS tasks, it can move from one L8 mode TCB to another. In this environment, the TCBS option returns the number of L8 mode TCBs that are using a DB2 connection at the time of the inquiry, and this value varies depending on workload.

THREADERROR (*cvda*)

Returns the processing that is to occur following a create thread error. CVDA values are as follows:

ABEND

For a second or subsequent SQL error, the transaction is abended with abend code AD2S, AD2T, or AD2U, depending on the type of error that occurred. The transaction must be stopped and reinitialized before it is allowed to issue another SQL request.

N906D

A transaction dump is to be taken and the DSNCSQL RMI associated with the transaction is *not* to be disabled. The transaction receives a -906 SQLCODE if another SQL is issued, unless the transaction issues SYNCPOINT ROLLBACK command. SYNCPOINT without the ROLLBACK option results in an ASP3 or ASP7 abend. The transaction dump records an abend of AD2S, AD2T, or AD2U.

N906

The DSNCSQL RMI associated with the transaction is *not* to be disabled. The transaction receives a -906 SQLCODE if another SQL request is issued, unless the transaction issues a SYNCPOINT ROLLBACK command. SYNCPOINT without the ROLLBACK option results in an ASP3 or ASP7 abend.

THREADLIMIT (*data-area*)

Returns the current maximum number of pool threads that the CICS DB2 attachment allows active before requests are made to wait or are rejected. See THREADWAIT.

THREADS (*data-area*)

Returns the current number of active pool threads.

THREADWAIT (*cvda*)

Returns a value showing whether transactions wait for a pool thread or are stopped if the number of active pool threads reaches the THREADLIMIT number. CVDA values are as follows:

TWAIT

If all threads are busy, a transaction waits until one becomes available.

NOTWAIT

If all threads are busy, a transaction is stopped with an abend code AD3T.

Conditions**NOTFND**

RESP2 values:

1 The DB2CONN cannot be found.

NOTAUTH

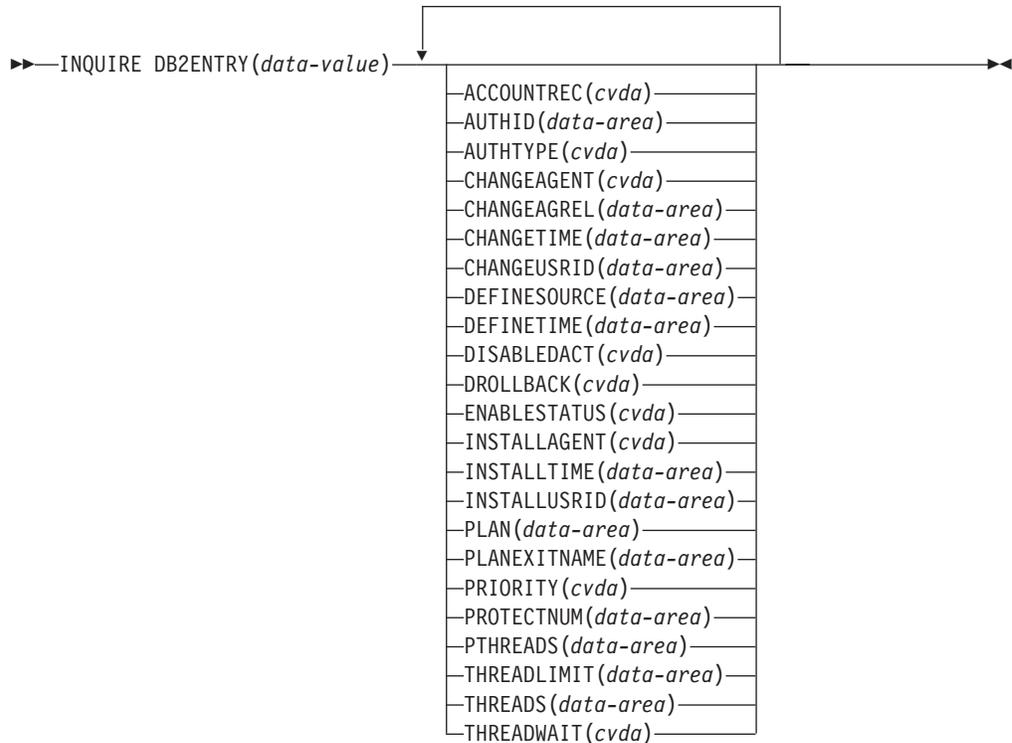
RESP2 values:

100 Command authorization failure

INQUIRE DB2ENTRY

Returns the attributes of the DB2ENTRY that defines resources to be used by a specific transaction or by a group of transactions when accessing DB2.

INQUIRE DB2ENTRY



Conditions: NOTAUTH, NOTFND

This command is threadsafe.

Description

The entry is identified by the name it was defined with in the CSD by the DEFINE DB2ENTRY command.

Browsing

You can also browse through all of the DB2ENTRY definitions installed in a CICS region by using the browse options, START, NEXT, and END, on INQUIRE DB2ENTRY commands. See "Browsing resource definitions" on page 20 for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

ACCOUNTREC

Returns the minimum amount of DB2 accounting required for transactions using this DB2ENTRY. The specified minimum can be exceeded, as described in the following options. CVDA values are as follows:

UOW The CICS DB2 attachment facility causes an accounting record to be produced by DB2 for each UOW, assuming that the thread is released at the end of the UOW.

TASK The CICS DB2 attachment facility causes a minimum of one accounting record to be produced by DB2 for each CICS task.

A transaction containing multiple UOWs can use a different thread for each UOW, assuming that the thread is released at sync point. The result can be the production of an accounting record for each UOW. For example, an accounting record is produced if a thread ends after being released, or if a thread is reused but the primary AUTHID is changed.

TXID The CICS DB2 attachment facility causes an accounting record to be produced by DB2 when the transid using the thread changes.

This option applies to DB2ENTRY definitions that are used by more than one transaction ID. Because threads are typically released at sync point, a transaction containing multiple UOWs can use a different thread for each UOW. The result can be that an accounting record is produced for each UOW. For example, an accounting record is produced if a thread stops after being released, or if a thread is reused but the primary AUTHID is changed.

NONE

No accounting records are required for transactions using threads from this DB2ENTRY.

DB2 produces, however, at least one accounting record per thread when the thread is ended. Additionally, authorization changes cause accounting records to be produced.

AUTHID

Returns an ID to be used for security checking for threads on this DB2ENTRY. If an AUTHID is returned, AUTHTYPE is not applicable.

AUTHTYPE

Returns the type of ID to be used for security checking for threads on this DB2ENTRY. If an AUTHTYPE is returned, AUTHID is blank. CVDA values are as follows:

GROUP

The 8-character user ID and the connected group name are used as the authorization ID. The following table shows how these two values are interpreted by DB2.

IDs passed to DB2	How DB2 interprets values
CICS sign-on user ID (USERID)	Represents the primary DB2 authorization ID.

IDs passed to DB2	How DB2 interprets values
RACF-connected group name	If the RACF list of group options is not active, then DB2 uses the connected group name supplied by the CICS attachment facility as the secondary DB2 authorization ID. If the RACF list of group options is active, DB2 ignores the connected group name supplied by the CICS attachment facility, but the value appears in the DB2 list of secondary DB2 authorization IDs.

If no RACF group ID is available for this user ID, an 8-character field of blanks is passed to DB2 as the group ID.

SIGN The SIGNID parameter of the DB2CONN is used as the resource authorization ID.

TERM The terminal identification (four characters padded to eight) is used as an authorization ID. An authorization ID cannot be obtained in this manner if a terminal is not connected with the transaction.

If a transaction is started using a CICS command and has no terminal associated with it, do not use AUTHTYPE(TERM).

TX The transaction identification (four characters padded to eight) is used as the authorization ID.

OPID The operator identification associated with the user ID that is associated with the CICS transaction is used as the authorization ID (three characters padded to eight).

USERID

The 8-character user ID associated with the CICS transaction is used as the authorization ID.

When the DB2 sample sign-on exit DSN3@.SGN is used with AUTHTYPE(USERID), the exit sends the user ID to DB2 as the primary authorization ID and the RACF group ID to DB2 as the secondary ID. When the sample sign-on exit is used, AUTHTYPE(USERID) and AUTHTYPE(GROUP) are the same.

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

| **CHANGETIME** (*data-area*)

| Returns an ABSTIME value that represents the time stamp when the resource
| definition was last changed. For more information about the format of the
| ABSTIME value, see FORMATTIME in CICS Application Programming.

| **CHANGEUSRID** (*data-area*)

| Returns the 8-character user ID that ran the change agent.

| **DEFINESOURCE** (*data-area*)

| Returns the 8-character source of the resource definition. The DEFINESOURCE
| value depends on the CHANGEAGENT value. For more information, see
| Summary of the resource signature field values in the Resource Definition
| Guide.

| **DEFINETIME** (*data-area*)

| Returns an ABSTIME value that represents the time stamp when the resource
| definition was created.

| **DISABLEDACT**

| Returns what CICS is to do with new transactions accessing the DB2ENTRY
| when it has been disabled or is disabling. If DISABLEDACT is not specified,
| and DB2ENTRY is disabled, new requests are routed to the pool by default.
| CVDA values are as follows:

| **POOL** The CICS DB2 attachment facility routes the request to the pool.
| Message DFHDB2072 is sent to the transient data destination specified
| by MSGQUEUE on the DB2CONN for each transaction routed to the
| pool.

| **ABEND**

| The CICS DB2 attachment facility stops the transaction. The abend
| code is AD26.

| **SQLCODE**

| An SQLCODE is returned to the application indicating that the
| DB2ENTRY is disabled.

| **DROLLBACK**

| Returns whether the CICS DB2 attachment initiates a sync point rollback if a
| transaction being selected as victim of a deadlock resolution. CVDA values are
| as follows:

| **ROLLBACK**

| The attachment facility issues a sync point rollback before returning
| control to the application. An SQL return code of -911 is returned to
| the application.

| **NOROLLBACK**

| The attachment facility is not to initiate a rollback for this transaction.
| An SQL return code of -913 is returned to the application.

| **ENABLESTATUS**

| Returns a cvda indicating whether the DB2ENTRY can be accessed by
| applications. CVDA values are as follows:

| **ENABLED**

| The DB2ENTRY can be accessed by applications. DB2ENTRY is
| installed in an ENABLED state.

| **DISABLED**

| The DB2ENTRY cannot be accessed by applications.

DISABLING

The DB2ENTRY is in the process of being disabled. New transactions cannot access the DB2ENTRY. Existing transactions using the DB2ENTRY are allowed to complete unless the DB2ENTRY is being disabled with the FORCE option.

INSTALLAGENT(*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

PLAN

Returns the name of the plan to be used for this DB2ENTRY. If PLAN is returned, PLANEXITNAME is blank.

PLANEXITNAME

Returns the name of the dynamic plan exit (if any) to be used for this DB2ENTRY. If PLANEXITname is returned, PLAN is blank.

PRIORITY

Returns the priority of the thread TCBs for this DB2ENTRY relative to the CICS main TCB (QR TCB). The thread TCBs are CICS open L8 TCBs. CVDA values are as follows:

HIGH Thread TCBs have a higher priority than the CICS QR TCB.

EQUAL

Thread TCBs have equal priority with the CICS QR TCB.

LOW Thread TCBs have a lower priority than the CICS QR TCB.

PROTECTNUM

Returns the maximum number of protected threads allowed for this DB2ENTRY.

PTHREADS

Returns the current number of protected threads for this DB2ENTRY. A protected thread is an inactive thread available for reuse by a new transaction. If no transaction has reused the thread by the time it has been processed by two purge cycles, the thread is ended.

THREADS

Returns the current number of threads active for this DB2ENTRY.

THREADLIMIT

Returns the current maximum number of threads for this DB2ENTRY that the CICS DB2 attachment allows active before requests are made to wait, overflow to the pool, or are rejected. See the THREADWAIT option.

THREADWAIT

Returns whether transactions wait for a DB2ENTRY thread be stopped, or to overflow to the pool if the number of active DB2ENTRY threads reaches the Threadlimit number. CVDA values are as follows:

TWAIT

If all threads are busy, a transaction waits until one becomes available.

NOTWAIT

If any threads are busy, a transaction is stopped with an abend code AD2P.

TPOOL

If all threads are busy, a transaction is diverted to use a pool thread. If the pool is also busy, and NOTWAIT has been specified for the THREADWAIT parameter on the DB2CONN, the transaction is stopped with an abend code AD3T.

Conditions

NOTAUTH

RESP2 values:

100 Command authorization failure

101 Resource authorization failure

NOTFND

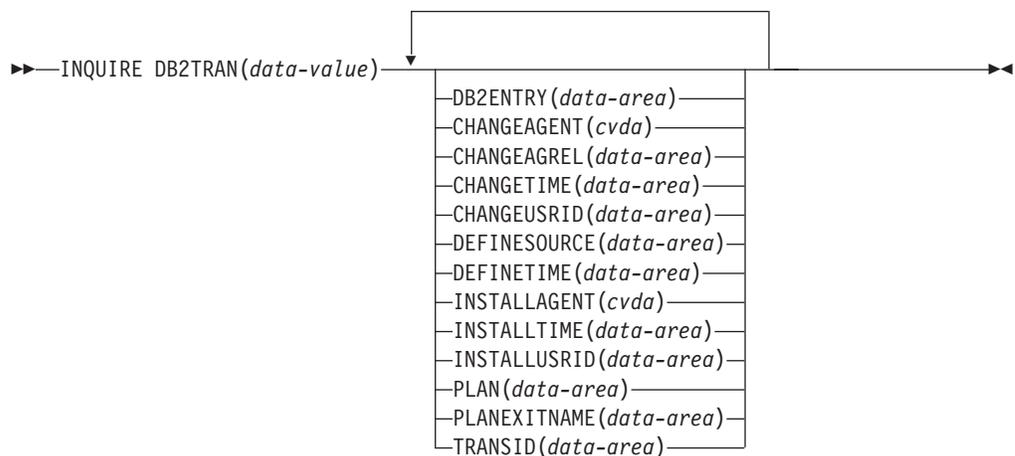
RESP2 values:

1 The DB2ENTRY cannot be found.

INQUIRE DB2TRAN

Returns attributes of a particular DB2TRAN definition; which associates a transaction or group of transactions with a DB2ENTRY.

INQUIRE DB2TRAN



Conditions: NOTAUTH, NOTFND

This command is threadsafe.

Description

A DB2TRAN is identified by the name with which it is defined in CEDA. Alternatively, if a TRANSID is specified on a DB2ENTRY when the DB2ENTRY is installed, CICS installs a DB2TRAN named DFHtttt, where tttt is the TRANSID.

Browsing

You can also browse through all of the DB2TRAN associations installed in your system by using the browse options, START, NEXT, and END, on INQUIRE DB2TRAN commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

DB2ENTRY

Returns the name of the DB2ENTRY to which this DB2TRAN refers; that is, the DB2ENTRY with which this additional transaction is associated.

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

DYNAMIC

The resource was defined as a result of the installation of a DB2ENTRY with TRANSID specified.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource

definition was last changed. For more information about the format of the ABSTIME value, see *FORMATTIME* in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The *DEFINESOURCE* value depends on the *CHANGEAGENT* value. For more information, see *Summary of the resource signature field values in the Resource Definition Guide*.

DEFINETIME (*data-area*)

Returns an *ABSTIME* value that represents the time stamp when the resource definition was created.

INSTALLAGENT (*cvda*)

Returns a *CVDA* value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a *CEDA* transaction or the programmable interface to *DFHEDAP*.

DYNAMIC

The resource was installed as a result of the installation of a *DB2ENTRY* with *TRANSID* specified.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME (*data-area*)

Returns an *ABSTIME* value that represents the time stamp when the resource was installed.

INSTALLUSRID (*data-area*)

Returns the 8-character user ID that installed the resource.

PLAN

Returns the name of the plan retrieved from the associated *DB2ENTRY* if it exists. If there is no associated *DB2ENTRY*, or if the *DB2ENTRY* is disabled with *DISABLEDACT(POOL)*, the pool plan name is returned if it exists. If *PLAN* is returned, *PLANEXITNAME* is blank.

PLANEXITNAME

Returns the name of the dynamic plan exit to be used, if any, from the associated *DB2ENTRY* if it exists. If there is no associated *DB2ENTRY*, or if the *DB2ENTRY* is disabled with *DISABLEDACT(POOL)*, the pool plan exit name is returned if it exists. If *PLANEXITNAME* is returned, *PLAN* is blank.

TRANSID

Specifies the transaction ID to be associated with the entry. The transaction ID can include wildcard characters. See the *CICS Resource Definition Guide* for information about use of wildcard characters.

Conditions

NOTAUTH

RESP2 values:

100 Command authorization failure

NOTFND

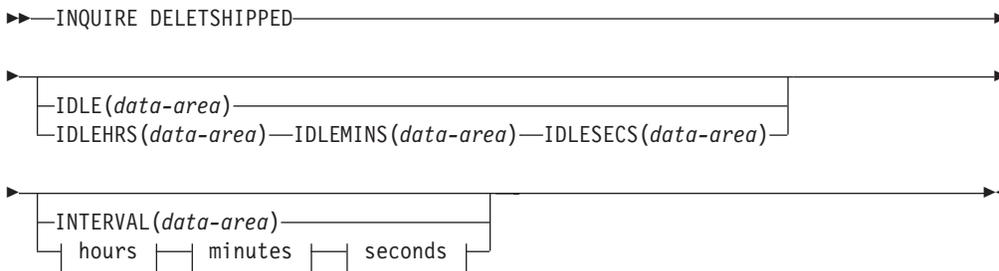
RESP2 values:

- 1 The DB2TRAN definition cannot be found.

INQUIRE DELETSHIPED

Retrieve information about system settings that control the CICS timeout delete mechanism.

INQUIRE DELETSHIPED



hours:



minutes:



seconds:



Conditions: NOTAUTH

Description

CICS provides a mechanism for deleting shipped terminal definitions after they have been idle for a period of time. The installation specifies how long a terminal must have been inactive to be eligible for deletion (the IDLE time), and how often the check should be made (the INTERVAL). The INQUIRE DELETSHIPED command displays the current settings of these two control options.

There are two formats for each of the time values that you can retrieve with this command (the idle time and the interval checking period):

- A 4-byte packed decimal composite (0hhmmss+), which you obtain by using the IDLE and INTERVAL options.
- Separate hours, minutes, and seconds, which you obtain by specifying the IDLEHRS, IDLEMINS, and IDLESECS options (instead of IDLE), and INTERVALHRS, INTERVALMINS, and INTERVALSECS (instead of INTERVAL).

Options

IDLE (*data-area*)

returns the idle time, as a 4-byte packed decimal field in the format 0hhmss+. Idle time is the minimum time that a terminal must be inactive to be eligible for deletion.

IDLEHRS (*data-area*)

returns the hours component of the idle time, in fullword binary form.

IDLEMIN (*data-area*)

returns the minutes component of the idle time, in fullword binary form.

IDLESECS (*data-area*)

returns the seconds component of the idle time, in fullword binary form.

INTERVAL (*data-area*)

returns a 4-byte packed decimal field, in the format 0hhmss+, giving the interval at which the check for idle terminals is made.

INTERVALHRS (*data-area*)

returns the hours component of the interval, in fullword binary form.

INTERVALMIN (*data-area*)

returns the minutes component of the interval, in fullword binary form.

INTERVALSECS (*data-area*)

returns the seconds component of the interval, in fullword binary form.

Conditions

NOTAUTH

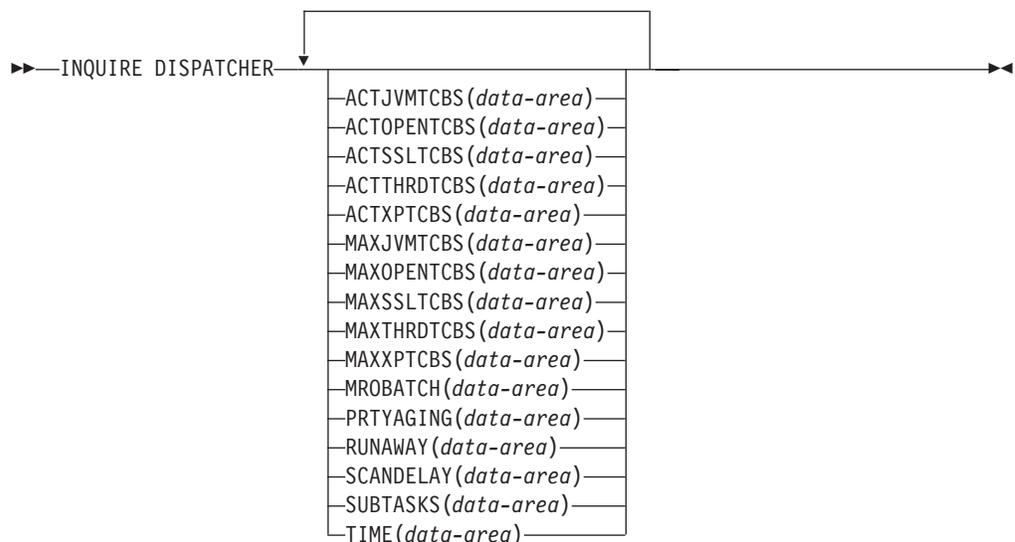
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

INQUIRE DISPATCHER

Retrieve CICS dispatcher information.

INQUIRE DISPATCHER



Conditions: NOTAUTH

For more information about the use of CVDAs, see CICS-value data areas (CVDAs) in CICS System Programming Reference.

Description

The **INQUIRE DISPATCHER** command returns information about the CICS system under which the task issuing the command is running.

Many of the options in this command correspond to options in the system initialization parameters and take their initial values from these parameters. You can change some of the parameter values by using the **SET DISPATCHER** command. Other options return CICS dispatcher state data.

Options

ACTJVMTCBS (*data-area*)

Returns a fullword binary field giving the number of J8 and J9 mode open TCBs currently allocated to user tasks.

The J8 and J9 mode TCBs are allocated from the pool of open TCBs that CICS attaches up to the maximum set by the **MAXJVMTCBS** system initialization parameter. CICS dispatcher maintains the pool of J8 and J9 mode TCBs for use only by Java programs that require a JVM. The ACTJVMTCBS value can be equal to, or less than, the MAXJVMTCBS value. If it is equal to MAXJVMTCBS, tasks that require a J8 or J9 mode open TCB are made to wait.

ACTOPENTCBS (*data-area*)

Returns a fullword binary field giving the total number of L8 and L9 mode open TCBs currently allocated to user tasks.

The L8 and L9 mode TCBs are allocated from the pool of open TCBs that CICS attaches up to the maximum set by the **MAXOPENTCBS** system initialization parameter. CICS dispatcher maintains the pool of L8 and L9 mode TCBs for use by OPENAPI applications and by task-related user exits that are enabled with the OPENAPI option. Task related user exits use only L8 mode TCBs; for example, the CICS DB2 adapter when connecting to DB2. The ACTOPENTCBS value can be equal to, or less than, the MAXOPENTCBS value. If it is equal to MAXOPENTCBS, tasks that require an L8 or L9 mode open TCB are made to wait.

ACTSSLTCBS (*data-area*)

Returns a fullword binary field giving the total number of S8 mode open TCBs currently allocated to user tasks.

The S8 mode TCBs are allocated from the pool of open TCBs that CICS attaches up to the maximum set by the **MAXSSLTCBS** system initialization parameter. S8 TCBs are used by tasks that require SSL functions. The ACTSSLTCBS value can be equal to, or less than, the MAXSSLTCBS value. If it is equal to MAXSSLTCBS, tasks that require an S8 TCB are made to wait.

ACTTHRTCBS (*data-area*)

Returns a fullword binary field giving the total number of T8 mode open TCBs currently allocated to the enabled JVM servers.

The T8 mode TCBs are allocated from a pool of open TCBs that CICS attaches up to the maximum set by the JVMSERVER resource definition. One pool is used by one JVM server. CICS dispatcher maintains the pools of T8 mode TCBs for use in the JVM server runtime environment.

ACTXPTCBS (*data-area*)

Returns a fullword binary field giving the total number of X8 and X9 mode open TCBs currently allocated to user tasks.

The X8 and X9 mode TCBs are allocated from the pool of open TCBs that CICS attaches up to the maximum set by the **MAXXPTCBS** system initialization parameter. CICS dispatcher maintains the pool of X8 and X9 mode TCBs for use by C and C++ programs compiled with the XPLINK option. The ACTXPTCBS value can be equal to, or less than, the MAXXPTCBS value. If it is equal to MAXXPTCBS, tasks that require an X8 or X9 mode open TCB are made to wait.

MAXJVMTCBS (*data-area*)

Returns a fullword binary field giving the maximum number of J8 and J9 mode open TCBs that CICS is allowed to attach and maintain in its pool of J8 and J9 mode TCBs. For information about the number allocated, see the ACTJVMTCBS option.

The difference between MAXJVMTCBS and ACTJVMTCBS represents the number of such TCBs that are free.

MAXOPENTCBS (*data-area*)

Returns a fullword binary field giving the maximum number of L8 and L9 mode open TCBs that CICS is allowed to attach and maintain in its pool of L8 and L9 mode TCBs. For information about the number allocated, see the ACTOPENTCBS option.

The difference between MAXOPENTCBS and ACTOPENTCBS represents the number of such TCBs that are free.

MAXSSLTCBS (*data-area*)

Returns a fullword binary field giving the maximum number of S8 mode open TCBs that CICS is allowed to attach and maintain in its pool of S8 mode TCBs.

MAXTHRDTCBS (*data-area*)

Returns a fullword binary field giving the maximum number of T8 mode open TCBs that CICS can attach and maintain for all JVM servers. CICS calculates the maximum number by adding one TCB for each JVM server to the value of the THREADLIMIT attribute from all the enabled and disabled JVMSERVER resources. The value of MAXTHRDTCBS cannot exceed 1024.

The difference between MAXTHRDTCBS and ACTTHRDTCBS represents the number of TCBs that are free. If you initialize another JVM server, one TCB is reserved for the JVM server.

MAXXPTCBS (*data-value*)

Returns a fullword binary field giving the maximum number of X8 and X9 mode open TCBs that CICS is allowed to attach and maintain in its pool of X8 and X9 mode TCBs. For information about the number allocated, see the ACTXPTCBS option.

The difference between MAXXPTCBS and ACTXPTCBS represents the number of such TCBs that are free.

MROBATCH (*data-area*)

Returns a fullword binary field giving the number of events that must occur, from a list of MRO and DASD I/O events on which CICS is waiting, before CICS is posted explicitly to process them.

PRTYAGING (*data-area*)

Returns a fullword binary field giving the rate at which CICS increases the

priority of a task waiting for dispatch. CICS increases the task priority by 1 after each PRTYAGING milliseconds of wait time without a dispatch.

RUNAWAY(*data-area*)

Returns a fullword binary field giving the default system value for runaway task time. This value is used for any task running a transaction with a profile that does not specify runaway task time. See the **INQUIRE TRANSACTION** option RUNAWAY.

SCANDELAY(*data-area*)

Returns a fullword binary field giving the maximum number of milliseconds between a user task making a terminal I/O request and CICS dispatching the terminal control task to process it. This value is sometimes called the “terminal scan delay”, and is set by the ICVTSD option in the system initialization table.

SUBTASKS*data-area*)

Returns a fullword binary field giving the value set by the **SUBTSKS** system initialization parameter, which can be either 0 or 1.

TIME(*data-area*)

Returns a fullword binary field giving the maximum interval in milliseconds for which CICS gives control to the operating system if no tasks are ready for dispatch. This value is set by the **ICV** system initialization parameter and is sometimes called the “region exit time interval”.

Conditions

NOTAUTH

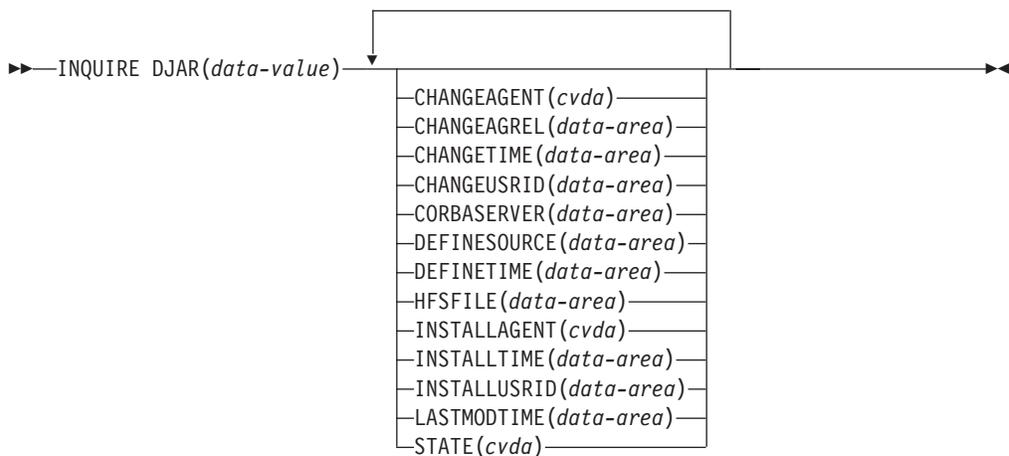
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

INQUIRE DJAR

Retrieve information about a DJAR definition.

INQUIRE DJAR



Conditions: NOTAUTH, NOTFND

For more information about the use of CVDAs, see “CICS-value data areas (CVDA)” on page 8.

Description

The INQUIRE DJAR command returns the attributes associated with a deployed JAR file.

Browsing

You can also browse through all the deployed JAR files installed in the specified CorbaServer, using the browse options (START, NEXT, and END) on INQUIRE DJAR commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT(*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

DYNAMIC

The resource was defined by the CICS scanning mechanism.

CHANGEAGREL(*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see FORMATTIME in CICS Application Programming.

|
|
|
|

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

CORBASERVER (*data-area*)

Specifies a 4-character area to receive the name of the associated CorbaServer.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

DJAR (*data-value*)

Specifies a 1- to 32-character deployed JAR file name.

HFSFILE (*data-area*)

Specifies a 255-character area to receive the fully qualified z/OS UNIX file name.

If this DJAR definition was created by performing the SCAN function on a CORBASERVER definition, this fully qualified z/OS UNIX file name is the one that appears in the DJARDIR parameter of that CORBASERVER definition.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

DYNAMIC

The resource was installed by the CICS scanning mechanism.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID (*data-area*)

Returns the 8-character user ID that installed the resource.

LASTMODTIME (*data-area*)

Specifies an 8-byte packed-decimal field to receive the time, in milliseconds since 00:00 on January 1st 1900, when the deployed JAR file on z/OS UNIX was last updated. This value is read-only, and CICS updates it when the DJAR resource is installed or updated. The last-modified time can be used to determine whether CICS has refreshed itself after an update is made to a JAR in the pickup directory.

For dynamically installed DJARs (those installed by the CICS scanning mechanism), the value of LASTMODTIME is the time stamp of the z/OS UNIX file to which the DJAR definition points, at the time the DJAR definition was last installed or updated.

For statically installed DJARs (those installed from a CSD or by CREATE DJAR), the value of LASTMODTIME is the time stamp of the z/OS UNIX file to which the DJAR definition points, at the time the DJAR was installed. The value is fixed; it cannot be updated like a dynamically-installed DJAR.

You can use the EXEC CICS FORMATTIME command to convert the LASTMODTIME value to the date-and-time format that you prefer.

If you issue an INQUIRE DJAR command before a newly installed or updated DJAR has fully initialized, the returned LASTMODTIME value will be zero.

STATE (*cvda*)

Specifies a CVDA value indicating the state of the deployed JAR file. The CVDA values are as follows:

DISCARDING

A DISCARD is in progress for this deployed JAR file.

INITING

The jar file is being copied to the shelf.

INSERVICE

Resolution of the copy of the jar file on the shelf has succeeded and the deployed JAR file is usable.

PENDINIT

Copying of the jar file to the shelf has not yet started.

PENDRESOLVE

Resolution of the copy of the jar file on the shelf has not yet started.

RESOLVING

The copy of the jar file on the shelf is being resolved.

UNRESOLVED

Resolution of the copy of the jar file on the shelf has failed and the deployed JAR file is unusable.

UNUSABLE

Copying the jar file to the shelf has failed and the deployed JAR file is unusable.

Conditions

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

NOTFND

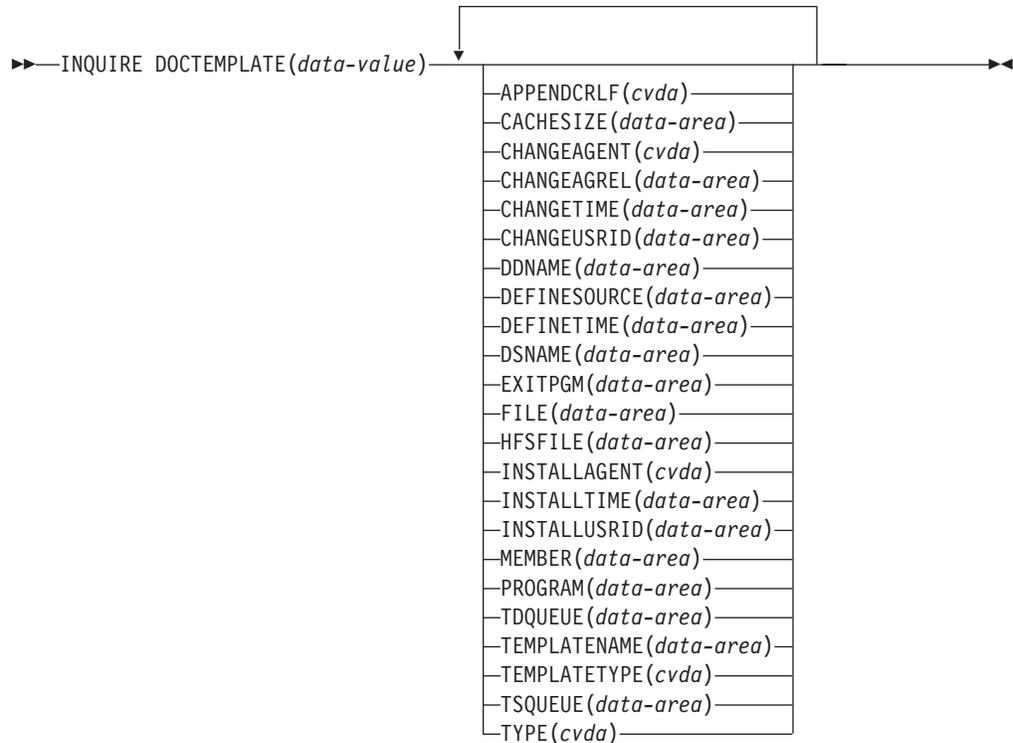
RESP2 values:

2 The named deployed JAR file was not found

INQUIRE DOCTEMPLATE

Find information about a DOCTEMPLATE resource definition for a CICS document template.

INQUIRE DOCTEMPLATE



Conditions: END, ILLOGIC, NOTFND, NOTAUTH

This command is threadsafe.

Description

Use the `INQUIRE DOCTEMPLATE` command to determine whether a particular `DOCTEMPLATE` resource definition is installed (defined in the current execution of your CICS system).

Browsing

You can also browse through all of the `DOCTEMPLATE` resource definitions installed in your system by using the browse options, `START`, `NEXT`, and `END`, on `INQUIRE DOCTEMPLATE` commands. See *Browsing resource definitions in CICS System Programming Reference* for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see *Auditing resources in the Resource Definition Guide*. The resource signature fields are `CHANGEAGENT`, `CHANGEAGREL`, `CHANGETIME`, `CHANGEUSRID`, `DEFINESOURCE`, `DEFINETIME`, `INSTALLAGENT`, `INSTALLTIME`, and `INSTALLUSRID`. See *Summary of the resource signature field values in the Resource Definition Guide* for detailed information about the content of the resource signature fields.

Options

APPENDCRLF (*cvda*)

Returns a CVDA value identifying whether CICS is to delete trailing blanks from and append carriage-return line-feed to each logical record of the template. CVDA values are as follows:

APPEND

Delete trailing blanks from and append carriage-return line-feed to each logical record of the template.

NOAPPEND

Do not delete trailing blanks from or append carriage-return line-feed to each logical record of the template.

CACHESIZE (*data-area*)

Returns a fullword binary field giving the amount of storage, in bytes, used by the cached copy of the document template. A value of zero is returned if there is no cached copy of the template at the time of the inquiry.

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

DYNAMIC

The resource was defined by the CICS system for a template being used through the CICS template manager, DFHWBTL.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

DDNAME (*data-area*)

Returns the 8-character DD name of the PDS containing the document template. The DD name applies only to a template of type PDS.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE**

value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

DOCTEMPLATE (*data-value*)

Specifies the 8-character identifier of the DOCTEMPLATE resource definition about which you are inquiring.

DSNAME (*data-area*)

Returns the 44-character data set name of the PDS containing the document template. It applies only to a template of type PDS.

EXITPGM (*data-area*)

Returns the 8-character name of the exit program to be called when a request is made for this document template. The exit program is passed an architected commarea containing the address and length of a buffer into which the exit program returns the template.

FILE (*data-area*)

Returns the 8-character name of the CICS file definition for the data set containing the document template.

HFSFILE (*data-area*)

Returns the fully qualified name of the z/OS UNIX System Services file where the document template resides. This name can be up to 255 characters in length.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

DYNAMIC

The resource was installed by the CICS system for a template being used through the CICS template manager, DFHWBTL.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID (*data-area*)

Returns the 8-character user ID that installed the resource.

MEMBER (*data-area*)

Returns the 8-character name of the member in the PDS containing the document template. MEMBER applies only to a template of type PDS.

PROGRAM(*data-area*)

Returns the 8-character name of the program in which the document template data is stored. CICS loads the program and takes all data after the entry point to be the template.

TDQUEUE(*data-area*)

Returns the 4-character name of the TD queue on which the document template is stored.

TEMPLATENAME(*data-area*)

Returns the extended template name by which the document template is to be known outside the resource definition function; that is, the TEMPLATENAME attribute of the DOCTEMPLATE resource definition. The name can be up to 48 characters long.

TEMPLATETYPE(*cvda*)

returns a CVDA value identifying the type of the source of this document template. CVDA values are as follows:

EXIT An exit program

FILE A CICS file name for a data set

HFSFILE

A z/OS UNIX System Services file

PDSMEMBER

A name of the member in the PDS described in DDNAME

PROGRAM

A program

TDQ A TD queue

TSQ A TS queue

TSQUEUE(*data-area*)

Returns the 16-character name of the TS queue on which the document template is stored.

TYPE(*data-area*)

Returns a CVDA value identifying the format of the template contents. CVDA values are as follows:

BINARY**EBCDIC****Conditions****END**

RESP2 values:

2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this DOCTEMPLATE resource definition in the way required by this command.

NOTFND

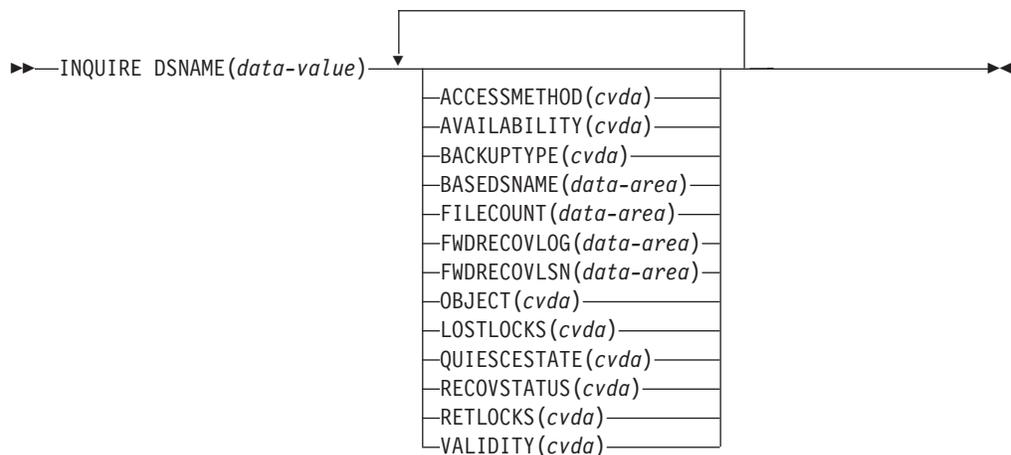
RESP2 values:

- 1 The DOCTEMPLATE specified cannot be found.

INQUIRE DSNAME

Retrieve information about an external data set.

INQUIRE DSNAME



Conditions: DSNNOTFOUND, END, ILLOGIC, IOERR, NOTAUTH

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The **INQUIRE DSNAME** command returns information about the object associated with a FILE resource definition, which can be a BDAM data set, a VSAM data set, or a VSAM path to a data set through an alternate index.

Data sets are associated with files either dynamically, through the DSNAME attribute in the FILE definition, or statically, through the DSN option on the associated JCL DD statement. Many of the attributes of a data set cannot be determined until the first file that references the data set has been opened by the CICS region in which the command is issued. Where an attribute is not valid until a file has been opened, the NOTAPPLIC state is returned.

Note: Using options that require a read from the ICF catalog can slow down the processing of this command.

Browsing

You can also browse through all the objects associated with files installed in your system, by using the browse options (START, NEXT, and END) on **INQUIRE DSNAME** commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

Options

ACCESSMETHOD(*cvda*)

Returns a CVDA value identifying the access method used with this data set. CVDA values are:

BDAM

The access method is BDAM.

NOTAPPLIC

The data set has not been opened by the CICS region in which the command is issued.

VSAM

The access method is VSAM.

AVAILABILITY(*cvda*) (**VSAM only**)

Returns a CVDA value indicating whether the data set is currently flagged, in this CICS region, as available or unavailable for use. The availability indicator is a local flag that a CICS region maintains in a data set name block (DSNB) for each data set. CVDA values are:

AVAILABLE

The data set is available for use according to the CICS data set name block. CICS can issue both RLS and non-RLS open requests for this data set.

Note: Although a data set is available according to information held by CICS, an open request could still fail if the ICF catalog indicates otherwise. This can occur, for example, if data set recovery is pending or in progress.

NOTAPPLIC

The data set is not a VSAM data set, or the data set has not been opened by the CICS region in which this command is issued.

UNAVAILABLE

Returned for a data set that CICS has marked as not available for use. The CICS region is unable to open the data set in either RLS or non-RLS mode.

BACKUPTYPE(*cvda*) (**VSAM only**)

Returns a CVDA value identifying the type of backup used for this data set. CVDA values are:

DYNAMIC

The data set is eligible for “backup while open” (BWO) processing; that is, a data set manager with the required function can take a backup of the data set while it is open for output. The data set can also be backed up while it is closed. The data set is eligible for BWO and it is accessed in non-RLS mode.

If the data set is opened in RLS mode, you need to look in the VSAM catalog to find out whether the data set is eligible for BWO. NOTAPPLIC is returned as the BACKUPTYPE for data sets opened in RLS mode.

NOTAPPLIC

The data set has not been opened by the CICS region in which the command is issued, or the data set is BDAM or a VSAM PATH. Also, if the data set has been opened in RLS mode, NOTAPPLIC is returned. The VSAM catalog should be referred to get the BWO status.

STATIC

The data set is accessed in non-RLS mode, and is not eligible for BWO processing. All CICS files open for output against this data set must be closed before a data set manager, such as DFSMSHsm or DFSMSdss, can take a backup copy. Hierarchical storage manager (DFSMSHsm) and data set services (DFSMSdss) are components of Data Facility Storage Management Subsystem (DFSMS/MVS).

If the data set is opened in RLS mode, you need to look in the VSAM catalog to find out whether the data set is eligible for BWO.

UNDETERMINED

Returned for base files if RECOVSTATUS is UNDETERMINED.

BASEDSNAME(*data-area*) (VSAM only)

Returns the 44-character name of the base cluster associated with a VSAM path, when the object of the inquiry is a path. When the object is a VSAM data set, this option returns the same value as the DSNNAME option.

Blanks are returned if the access method is BDAM, or if the data set has not been opened by the CICS region in which the command is issued.

DSNAME(*data-value*)

Specifies the 44-character identifier of the object about which you are inquiring. It must be associated with a FILE definition installed in CICS, named either in the DSNNAME option of that definition or the JCL DD statement specified in the DDNAME option.

FILECOUNT(*data-area*)

Returns a fullword binary field indicating the number of installed file definitions that refer to this data set.

FWDRECOVLOG(*data-area*) (VSAM only)

Returns, as a half-word binary value, the numeric journal identifier of the journal being used as the forward-recovery log, if this is a forward-recoverable data set.

FWDRECOVLOG is undefined if the data set is not forward-recoverable. A data set can be defined as being forward recoverable in the ICF catalog or, if it is accessed in non-RLS mode, in the file definition.

This option is valid for data sets accessed only in non-RLS mode, and for which the recovery attributes are obtained from the file resource definition.

CICS returns a value of zero for forward-recoverable data sets accessed in RLS mode, or for non-RLS mode data sets for which CICS obtains the recovery attribute from the ICF catalog.

FWDRECOVLSN(*data-area*) (VSAM only)

Returns the name (up to 26 characters) of the log stream that is used to log the

updates if this is a data set defined with forward-recovery attributes. CICS returns blanks if the data set is not forward recoverable.

The log stream name returned is either:

- The log stream name specified directly in the ICF catalog for DFSMS/MVS 1.3 or above
- For a non-RLS access mode data set that does not have forward recovery attributes in the ICF catalog, it is a log stream name identified by CICS through a journal name generated from the FWDRECOVLOG value.

LOSTLOCKS(*cvda*) (RLS only)

Returns a CVDA value indicating whether there are any lost locks for this data set. CVDA values are:

NOTAPPLIC

This is not an RLS data set, or the data set has not been opened by the CICS region in which the command is issued.

NOLOSTLOCKS

The data set has no lost locks.

REMLOSTLOCKS

The data set has lost locks, hence is unavailable, but no recovery is required on this CICS region.

RECOVERLOCKS

The data set has lost locks, hence is unavailable, and the CICS region is performing lost-locks recovery.

See the RESETLOCKS and the FORCE|COMMIT|BACKOUT options on the **EXEC CICS SET DSNAME** command for information about purging units of work that might be holding up lost locks recovery.

OBJECT(*cvda*) (VSAM only)

Returns a CVDA value indicating whether the object of the inquiry is a real data set containing records (a VSAM KSDS, ESDS, or RRDS, or an alternate index used directly) or a VSAM path definition that links an alternate index to its base cluster. CVDA values are:

BASE This is a data set containing records.

NOTAPPLIC

The data set has not been opened by the CICS region in which the command is issued, or it is a BDAM data set.

PATH This is a path.

QUIESCESTATE(*cvda*) (VSAM only)

Returns a CVDA value indicating the RLS quiesce state of the data set. The information is obtained from the ICF catalog entry for the data set.

Note: This option is returned, whether or not the data set has been opened by the CICS region in which the command is issued.

CVDA values are:

NOTAPPLIC

This data set is:

- Migrated
- Accessed using BDAM
- Accessed using a level of VSAM that does not support RLS (that is, DFSMS/MVS is earlier than 1.3)

NOTAPPLIC is also returned if CICS is running without RLS support (the RLS=NO system initialization parameter is specified or implied).

QUIESCED

This data set has been quiesced. CICS cannot open files in RLS mode against the data set, and no CICS region has a file currently open against this data set. However, the data set can be opened in non-RLS mode.

QUIESCING

This data set is in the process of quiescing. It applies only to the CICS region that initiated the quiesce; for other CICS regions, UNQUIESCED is returned.

UNQUIESCED

The normal value for a data set that is not quiescing or is not quiesced. It indicates that files can be opened in RLS or non-RLS mode against the data set, the mode being established by the first open. After a file is opened in one mode, other files can be opened only in the same mode.

RECOVSTATUS(*cvda*)

Returns a CVDA value identifying the recovery characteristics of the data set. CVDA values are:

FWDRECOVERABLE

All updates to the data set are logged for both backout and forward recovery.

NOTAPPLIC

This is a BDAM data set or a VSAM path, or the data set has not been opened by the CICS region in which the command is issued.

NOTRECOVERABLE

Updates to the data set are not logged.

This response may also be returned as the result of use of the XFCNREC global user exit. A program enabled at XFCNREC may indicate that file opens should proceed even if there is a mismatch in the backout recovery requirements for different files associated with same data set. In these circumstances, the data set is marked as NOTRECOVERABLE to indicate that its data integrity can no longer be guaranteed. The condition remains until cleared by a **CEMT SET DSNAME REMOVE** or **EXEC CICS SET DSNAME REMOVE** command, or by an initial or cold start.

While the data set is in this state, backout logging is performed for a particular request based on the specification in the file definition. Therefore backout logging may occur for requests via one file and not via another.

RECOVERABLE

All updates to the data set are logged for backout.

UNDETERMINED

The recovery status is unknown because no files associated with this data set have been opened, or because the only files opened were defined as coupling facility data tables or as user-maintained data tables (where the recovery attributes are independent of the associated data set).

RETLOCKS (*cvda*)

Returns a CVDA value indicating whether there are any retained record locks, as a result of deferred recovery work by this CICS region, for the specified data set. CVDA values are:

NOTAPPLIC

This data set has not been opened by the CICS region in which the command is issued.

NORETAINED

This CICS region:

- Has no deferred recovery work for the base data set, and therefore no retained locks, or
- Has recovery work currently in progress.

Retained locks might be held against the data set by other CICS regions. The command needs to be issued on all regions in the sysplex to get a full picture of the state of the data set. See the *CICS Operations and Utilities Guide* for information about the CICS batch-enabling sample programs that assist you in doing this, and about the AMS SHCDS LIST subcommands that allow you to investigate retained locks held by CICS regions that are down.

RETAINED

This CICS region has deferred recovery work causing retained locks for the data set. One effect of this is that, if the data set was last opened in RLS mode, the locks are RLS locks and, therefore, the data set cannot be opened in non-RLS mode.

Another effect is that any FILE definitions that specify this data set cannot be changed to specify a different data set.

If the data set is a BDAM data set, or a VSAM data set accessed in non-RLS mode, the locks are CICS record locks, otherwise they are RLS record locks. The UOW that has retained locks is usually shunted, but it may be in the process of being retried.

VALIDITY (*cvda*)

Returns a CVDA value identifying whether the data set name has been validated against the VSAM catalog by opening a file associated with the data set. CVDA values are:

INVALID

The data set name has not been validated (validation has not yet occurred or has failed).

VALID

The data set name has been validated.

You cannot find out what the RECOVSTATUS of a data set is unless VALIDITY has a setting of VALID.

Conditions**DSNNOTFOUND**

RESP2 values:

- 1 The data set cannot be found.

END

RESP2 values:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

IOERR

RESP2 values:

- 40 QUIESCESTATE was specified, but an error was raised by DFSMS/MVS when reading the ICF catalog.
- 48 The specified operation cannot be completed because the data set is migrated. Recall the data set and reissue the command.
- 49 An error was raised by DFSMS/MVS when reading the ICF catalog to establish the base data set name.

Note: If an IOERR occurs within a browse it does not terminate the browse operation, and CICS attempts to return as many parameter values as possible.

NOTAUTH

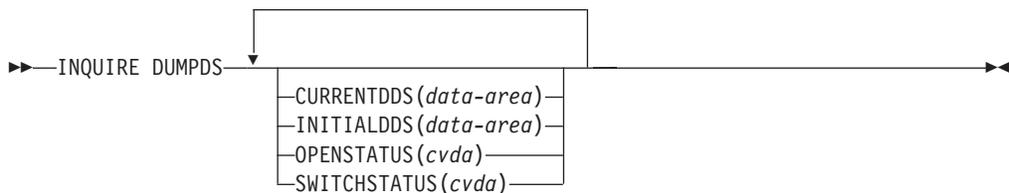
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

INQUIRE DUMPDS

Retrieve information about the CICS transaction dump data sets.

INQUIRE DUMPDS



Conditions: NOTAUTH

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The INQUIRE DUMPDS command allows you to retrieve information about CICS transaction dump data sets. There can either be one of these, known as the ‘A’ data set, or two: ‘A’ and ‘B’. One is “active” (receiving dumps) and the other, if there are two, is “inactive” (standby).

Options

CURRENTDDS (*data-area*)

returns the 1-character designator of the active dump data set (A or B). The active dump data set is not necessarily open.

INITIALDDS (*data-area*)

returns a 1-character value indicating which dump data set CICS designates as active at startup.

A Dump data set A is active initially.

B Dump data set B is active initially.

X The dump data set that was not active when CICS last terminated (normally or abnormally) is active initially.

OPENSTATUS (*cvda*)

returns a CVDA value identifying the status of the active CICS dump data set. CVDA values are:

CLOSED

The active CICS dump data set is closed.

OPEN The active CICS dump data set is open.

SWITCHSTATUS (*cvda*)

returns a CVDA value indicating whether CICS should switch active data sets when the current one fills. CVDA values are:

NOSWITCH

No automatic switching occurs.

SWITCHNEXT

When the data set designated as active at startup fills, CICS closes it, opens the other, and makes that one active. This automatic switch occurs only once, when the first active data set fills; thereafter, switching is under manual or program control.

Conditions

NOTAUTH

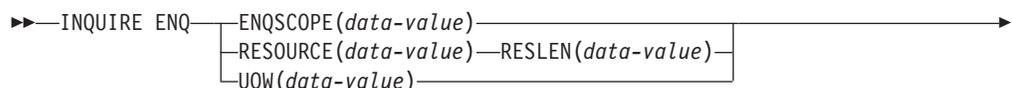
RESP2 values:

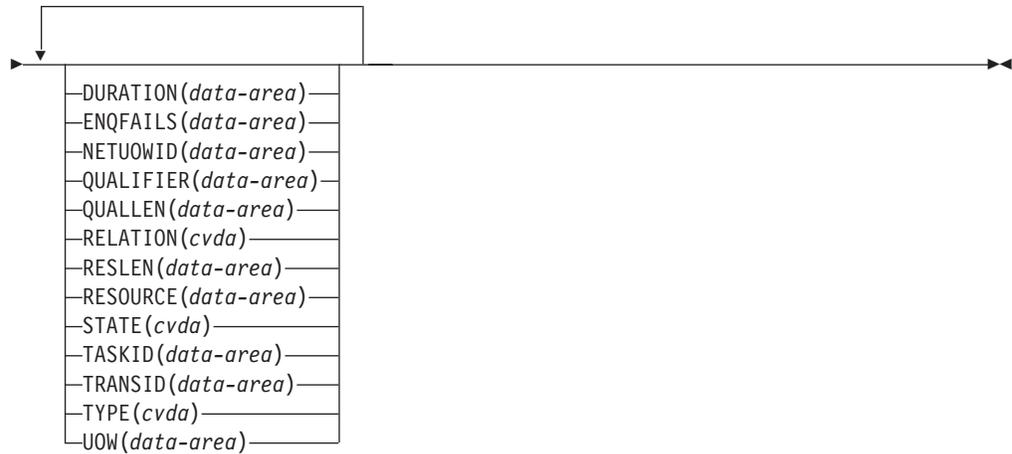
100 The user associated with the issuing task is not authorized to use this command.

INQUIRE ENQ

Retrieve information about enqueues held or waited on by a UOW, or about UOWs holding or waiting on a specified enqueue. INQUIRE ENQ is a synonym for INQUIRE UOWENQ; see "INQUIRE UOWENQ" on page 544 for a full description.

INQUIRE ENQ





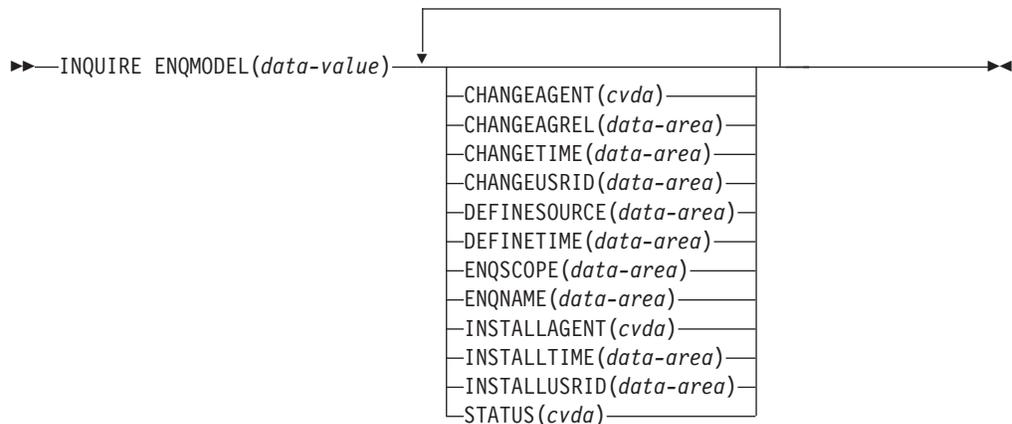
Conditions: END, ILLOGIC, NOTAUTH, UOWNOTFOUND

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

INQUIRE ENQMODEL

Retrieve information about enqueue model definitions on the local system.

INQUIRE ENQMODEL



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The INQUIRE ENQMODEL command returns information about enqueue model definitions on the local system.

You can make an explicit INQUIRE for a given ENQMODEL or use the browse form of the command. Browse returns all enqueue model definitions on the local system.

Browsing

To browse through all of the ENQ models in your local system, use the browse options (START, NEXT, and END) on INQUIRE ENQMODEL commands.

See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT(*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

SYSTEM

The resource definition was last changed by the CICS or CICSplex system.

CHANGEAGREL(*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see FORMATTIME in CICS Application Programming.

CHANGEUSRID(*data-area*)

Returns the 8-character user ID that ran the change agent.

| **DEFINESOURCE**(*data-area*)

| Returns the 8-character source of the resource definition. The **DEFINESOURCE**
| value depends on the **CHANGEAGENT** value. For more information, see
| Summary of the resource signature field values in the Resource Definition
| Guide.

| **DEFINETIME**(*data-area*)

| Returns an **ABSTIME** value that represents the time stamp when the resource
| definition was created.

| **ENQMODEL**(*data-value*)

| Specifies the 8-character identifier of an enqueue model.

| **ENQSCOPE**(*data-area*)

| Returns the 4-character name that qualifies sysplex-wide **ENQUEUE** requests
| issued by this CICS region. Four blanks indicate that the enqueue is **LOCAL**.

| **ENQNAME**(*data-area*)

| Returns the 1- to 255-character resource name or generic name.

| **ENQ** commands issued by this CICS region are checked against this resource
| or generic name. If a match is found, and **ENQSCOPE** was specified, the
| enqueue is sysplex-wide, qualified by the 4-character **ENQSCOPE**.

| **INSTALLAGENT**(*cvda*)

| Returns a **CVDA** value that identifies the agent that installed the resource. The
| possible values are as follows:

| **CREATESPI**

| The resource was installed by an **EXEC CICS CREATE** command.

| **CSDAPI**

| The resource was installed by a **CEDA** transaction or the
| programmable interface to **DFHEDAP**.

| **GRPLIST**

| The resource was installed by **GRPLIST INSTALL**.

| **INSTALLTIME**(*data-area*)

| Returns an **ABSTIME** value that represents the time stamp when the resource
| was installed.

| **INSTALLUSRID**(*data-area*)

| Returns the 8-character user ID that installed the resource.

| **STATUS**(*cvda*)

| Returns a **CVDA** value describing the current state of the **ENQMODEL**. **CVDA**
| values are as follows:

| **ENABLED**

| Matching enqueue requests are being processed in the normal way.

| **DISABLED**

| Matching enqueue requests are being rejected, and the issuing tasks are
| abending with code **ANQE**. Matching **INSTALL CREATE** or **DISCARD**
| requests are being processed.

| **WAITING**

| Matching enqueue requests are being rejected, and the issuing tasks are
| abending with code **ANQE**. **INSTALL CREATE** or **DISCARD** requests
| are waiting to be processed.

Conditions

END

RESP2 values:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

NOTFND

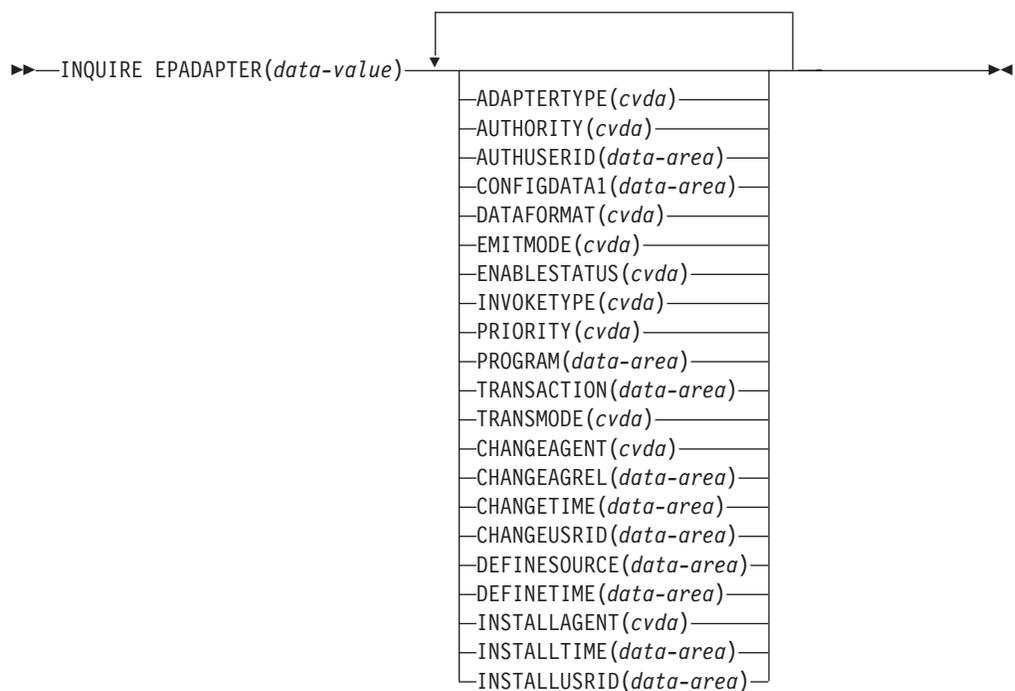
RESP2 values:

- 1 The ENQMODEL cannot be found.

INQUIRE EPADAPTER

Retrieve information about a specified event processing adapter.

INQUIRE EPADAPTER



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.

Description

The INQUIRE EPADAPTER command returns information about a specified event processing adapter.

Browsing

You can browse through all the event processing adapters that are installed in your region by using the browse options (START, NEXT, and END) on **INQUIRE EPADAPTER** commands.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

ADAPTERTYPE (*cvda*)

Returns a CVDA value indicating the type of this EP adapter. The CVDA values are as follows:

CUSTOM

A user-written EP adapter that emits events in any format that you require to any destination.

HTTP The HTTP EP adapter that emits events to an HTTP server for consumption by products such as WebSphere Business Events and IBM Business Monitor.

TRANSTART

The Transaction Start EP adapter that emits events to a named CICS transaction.

TSQUEUE

The TSQ EP adapter that emits events to a named CICS TS queue.

WMQ The WebSphere MQ EP adapter that emits events to WebSphere MQ for consumption by products such as WebSphere Business Events and IBM Business Monitor.

AUTHORITY (*cvda*)

Returns a CVDA value indicating the authority of the EP adapter. The CVDA values are as follows:

CONTEXT

The EP adapter runs using the user ID of the task that caused the event to be captured. This is always the case when EMITMODE is SYNCHRONOUS or when **Use Context User ID** is specified in the advanced section of the Adapter tab for the EP adapter.

|
| **REGION**

| The EP adapter runs using the CICS region user ID.

| **USERID**

| The EP adapter is attached using the identifier specified in the EP
| adapter user ID and returned in the AUTHUSERID attribute.

| **AUTHUSERID**(*data-area*)

| Returns the 8-character identifier to be used to attach the EP adapter
| transaction. This attribute is only set if AUTHORITY is USERID.

| **CHANGEAGENT**(*cvda*)

| Returns a CVDA value that identifies the agent that made the last change to
| the resource definition. The possible values are as follows:

| **CREATESPI**

| The resource definition was last changed by an **EXEC CICS CREATE**
| command.

| **CSDAPI**

| The resource definition was last changed by a CEDA transaction or the
| programmable interface to DFHEDAP.

| **CSDBATCH**

| The resource definition was last changed by a DFHCSDUP job.

| **DREPAPI**

| The resource definition was last changed by a CICSplex SM BAS API
| command.

| **CHANGEAGREL**(*data-area*)

| Returns a 4-digit number of the CICS release that was running when the
| resource definition was last changed.

| **CHANGETIME**(*data-area*)

| Returns an ABSTIME value that represents the time stamp when the resource
| definition was last changed. For more information about the format of the
| ABSTIME value, see FORMATTIME in CICS Application Programming.

| **CHANGEUSRID**(*data-area*)

| Returns the 8-character user ID that ran the change agent.

| **CONFIGDATA1**(*data-area*)

| A 64-character data area containing the primary configuration data item for the
| EP adapter. If the primary configuration data item is less than 64 bytes, the
| field is padded with blanks. The data item returned is dependent upon
| ADAPTERTYPE, as follows:

| **CUSTOM**

| Returns the first 64 bytes of custom EP adapter configuration data.

| **HTTP** Returns the 8-character name of the URIMAP definition to be used by
| an HTTP EP adapter to locate the HTTP server.

| **TRANSTART**

| Returns the 4-character name of the event consumer transaction that is
| started by a Transaction Start EP adapter.

| **TSQUEUE**

| Returns the 16-character name of the temporary storage queue for the
| event emitted by a TSQ EP adapter.

| **WMQ** Returns the 48-character name of the WebSphere MQ queue for event

messages emitted by this WebSphere MQ EP adapter. This data is in the code page defined by the **LOCALCCSID** system initialization parameter.

DATAFORMAT (*cvda*)

Returns a CVDA value indicating the format of events emitted by this EP adapter. The CVDA values are as follows:

USER The format is user-defined.

CBER Common Base Event REST format for the IBM Business Monitor REST HTTP server.

CBE Common Base Event format for consumption by products such as IBM Business Monitor.

CCE CICS Container Event format.

CFE CICS Flattened Event format.

WBE WebSphere Business Events XML format.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT** value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an **ABSTIME** value that represents the time stamp when the resource definition was created.

EMITMODE (*cvda*)

Returns a CVDA value indicating the event emission attribute of this EP adapter. The CVDA values are as follows:

ASYNCHRONOUS

Event emission and the capturing transaction are asynchronous. Failure to emit an event has no effect on the capturing transaction.

SYNCHRONOUS

Event emission is synchronous with the capturing transaction. The unit of work for the capturing transaction does not complete successfully when the event is not emitted.

ENABLESTATUS (*cvda*)

Returns a CVDA value indicating the status of this EP adapter. CVDA values are as follows:

ENABLED

The EP adapter is enabled.

DISABLED

The EP adapter is disabled.

EPADAPTER (*data-area*)

Specifies the name (1 - 32 characters) of an EP adapter. You must specify this option to retrieve details of a particular EP adapter by name. On the browse form of this command, you must provide a 32-character data area to receive the name of the EP adapter.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. Only one value is possible:

| **BUNDLE**

| The resource was installed by a bundle deployment.

| **INSTALLTIME**(*data-area*)

| Returns an ABSTIME value that represents the time stamp when the resource
| was installed.

| **INSTALLUSRID**(*data-area*)

| Returns the 8-character user ID that installed the resource.

| **INVOKETYPE**(*cvda*)

| Returns a CVDA value indicating how the EP adapter is started. The CVDA
| values are as follows:

| **ATTACH**

| The EP adapter is attached as a separate task.

| **LINK** The EP adapter program is linked to.

| **PRIORITY**(*cvda*)

| Returns a CVDA value indicating the dispatching priority of event emission for
| this EP adapter. This attribute is ignored when EMITMODE is
| SYNCHRONOUS. The CVDA values are as follows:

| **HIGH** Events emitted for this EP adapter are high priority.

| **NORMAL**

| Events emitted for this EP adapter are normal priority.

| **PROGRAM**(*data-value*)

| Returns the 8-character name of the EP adapter program. If the
| ADAPTERTYPE is CUSTOM, this attribute is only applicable if INVOKETYPE
| is LINK.

| **TRANSACTION**(*data-value*)

| Returns the 4-character name of the transaction definition that is used if the EP
| adapter transaction is attached. The TRANSACTION attribute is only
| applicable if INVOKETYPE is ATTACH.

| **TRANSMODE**(*cvda*)

| Returns a CVDA value indicating the event transactionality attribute of this EP
| adapter. CVDA values are as follows:

| **NONTRANS**

| Events are not transactional. Events can be emitted regardless of
| whether the unit of work for the capturing transaction completes
| successfully.

| **TRANS**

| Events are transactional. Events can be emitted only when the unit of
| work for the capturing transaction completes successfully.

| **Conditions**

| **END**

| RESP2 values:

| **2** There are no more EP adapters to browse.

| **ILLOGIC**

| RESP2 values:

| **1** You have issued a **START** command when a browse of this resource

type is already in progress, or you have issued a **NEXT** or an **END** command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

101 The user associated with the issuing task is not authorized to read the EP adapter.

NOTFND

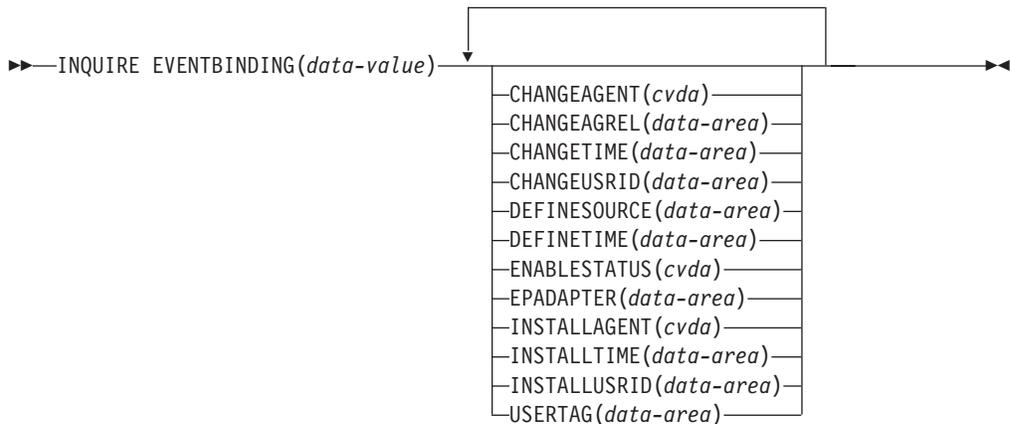
RESP2 values:

3 The specified EP adapter cannot be found.

INQUIRE EVENTBINDING

Retrieve information about a specified event binding.

INQUIRE EVENTBINDING



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.

Description

The **INQUIRE EVENTBINDING** command returns information about a specific event binding.

Browsing

You can browse through all the event bindings that are installed in your region by using the browse options (START, NEXT, and END) on **INQUIRE EVENTBINDING** commands.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources

in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCS DUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see FORMATTIME in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

ENABLESTATUS (*cvda*)

Returns a CVDA value indicating the status of this event binding. CVDA values are as follows:

ENABLED

The event binding is enabled.

DISABLED

The event binding is disabled.

EPADAPTER (*data-area*)

Returns the 32-character name of the EP adapter used by this event binding.

EVENTBINDING(*data-value*)

Specifies the 32-character name of an event binding.

INSTALLAGENT(*cvda*)

Returns a CVDA value that identifies the agent that installed the resource.
Only one value is possible:

BUNDLE

The resource was installed by a bundle deployment.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

USERTAG (*data-area*)

Returns the 8-character user tag of the event binding.

Conditions**END**

RESP2 values:

- 2 There are no more event bindings to browse.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to read the event binding.

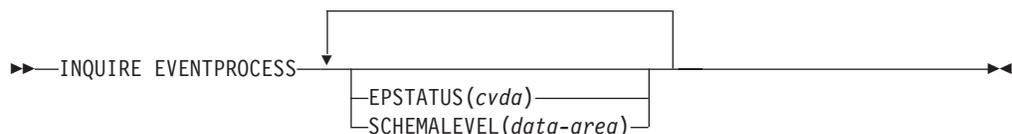
NOTFND

RESP2 values:

- 3 The specified event binding cannot be found.

INQUIRE EVENTPROCESS

Retrieve the status of event processing.

INQUIRE EVENTPROCESS

Conditions: NOTAUTH

This command is threadsafe.

Description

The INQUIRE EVENTPROCESS command returns the status of event processing.

Options

EPSTATUS(*cvda*)

Returns a CVDA value identifying the current status of event processing.

STARTED

CICS is processing events.

DRAINING

CICS event processing is draining.

STOPPED

CICS is not processing events.

SCHEMALEVEL(*data-area*)

Returns a 4-character value (*vvrr*) indicating the highest version and release of event binding schema that is supported by CICS, where *vv* is the version and *rr* is the release; for example, 0201 indicates version 2 release 1 of the event binding schema.

Conditions

NOTAUTH

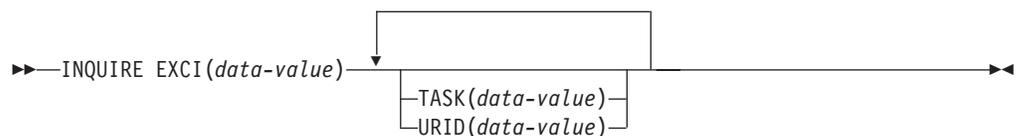
RESP2 values are:

100 The user associated with the issuing task is not authorized to use this command.

INQUIRE EXCI

Retrieve information about jobs using the external CICS interface.

INQUIRE EXCI



Conditions: END, ILLOGIC, NOTAUTH

Description

The **INQUIRE EXCI** command identifies the names of batch jobs currently connected to CICS through the interregion communication (IRC) facility.

Options

EXCI(*data-value*)

returns a 35-character string identifying the EXCI client job and on what z/OS system it is running .

TASK(*data-value*)

specifies, the fullword binary task number of the mirror transaction running on behalf of a specific batch job.

Information about jobs using the external CICS interface is available only after that job has issued at least one DPL request. A nonzero task number indicates that a DPL request is currently active. A zero task number indicates that an external CICS interface session is still open (connected) for that job, although no DPL request is currently active.

URID(*data-value*)

specifies, when the job is using RRMS to coordinate updates, and when there is an active DPL request for the session, a 32-character string containing the hexadecimal representation of the RRMS Unit of Recovery Identifier.

Conditions**END**

RESP2 values:

2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

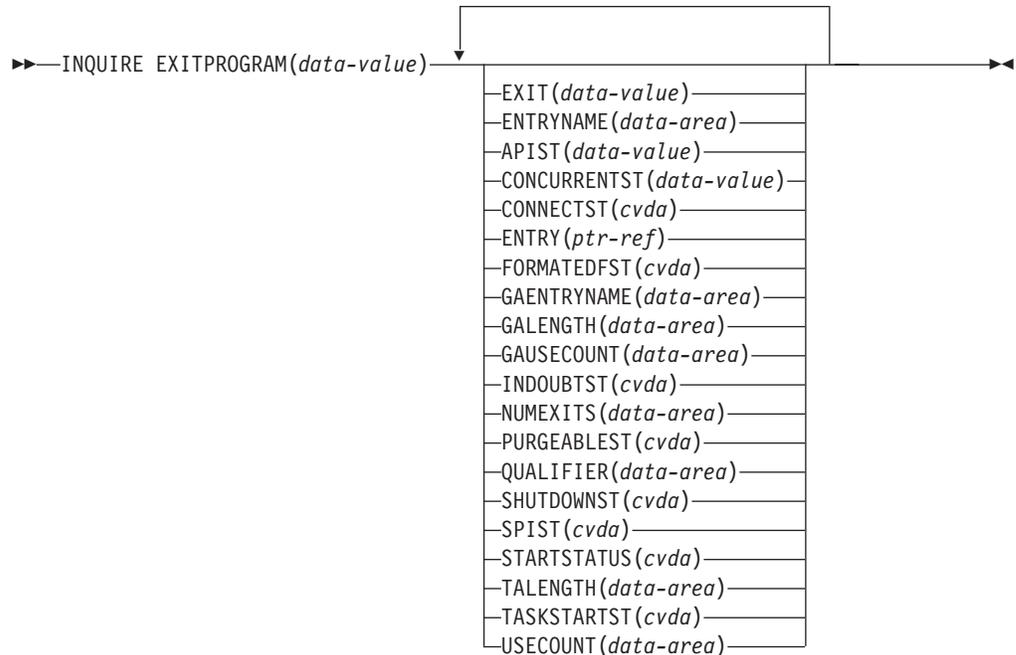
100 The user associated with the issuing task is not authorized to use this command.

101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

INQUIRE EXITPROGRAM

Retrieve information about a user exit.

INQUIRE EXITPROGRAM



Conditions: END, ILLOGIC, INVREQ, NOTAUTH, PGMIDERR

This command is threadsafe.

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The INQUIRE EXITPROGRAM command returns information about a global or task-related user exit. You identify the exit about which you are inquiring with the ENTRYNAME and EXITPROGRAM options.

Browsing

You can also browse through the exit definitions in two different ways. To look at all of the global user exits defined at a particular exit point, you specify the exit point on the command that starts the browse, thus:

```
INQUIRE EXITPROGRAM EXIT(data-value) START
```

To look at all user exits, both global and task-related, you omit the EXIT option on the command that starts the browse. You can distinguish between the two types by looking at the NUMEXITS value, which is zero for a task-related exit and positive for a global exit.

On either type of browse, the sequence in which the exits are retrieved is the time order in which they were enabled.

Options

APIST

returns a CVDA indicating which APIs the user exit program uses.

CVDA values are:

BASEAPI

CICSAPI has replaced BASEAPI. Both these CVDA values have the same meaning, and, for compatibility, BASEAPI is still accepted by the translator.

CICSAPI

The user exit program is enabled as either QUASIRENT or THREADSAFE, but without the OPENAPI option. This means it is restricted to the CICS permitted programming interfaces. In releases before CICS TS for z/OS, Version 3, this CVDA was called BASEAPI.

OPENAPI

The task-related user exit program is enabled with the OPENAPI option. This means it is permitted to use non-CICS API, for which purpose CICS will give control to the task-related user exit under an L8 mode open TCB. OPENAPI assumes that the program is written to threadsafe standards.

CONCURRENTST

returns a CVDA indicating the concurrency status of the global or task-related user exit program. This is the value of the CONCURRENCY attribute of the PROGRAM definition, or of any override specified by the latest ENABLE command for this program.

CVDA values are:

QUASIRENT

The exit program is defined as being quasi-reentrant, and is able to run only under the CICS QR TCB when invoking CICS services through the CICS API. To use any MVS services, a task-related user exit program must switch to a privately-managed TCB.

THREADSAFE

The exit program is defined as threadsafe, and is capable of running under an open TCB.

For task-related user exit programs only, if the APIST option returns OPENAPI the program will always be invoked under an open TCB.

For both global and task-related user exit programs, an APIST option of CICSAPI means that the program is invoked under whichever TCB is in use by its user task when the program is given control. This could be either an L8 mode open TCB or the CICS QR TCB.

CONNECTST(*cvda*) (task-related user exits only)

returns a CVDA value indicating the state of the connection between the exit and the external resource manager that it supports. CONNECTST enables you to determine whether the specified exit has connected to its resource manager, so that CICS tasks can safely issue API requests to the resource manager.

For example, to inquire about the connection to DBCTL, use an EXITPROGRAM value of DFHDBAT and an ENTRYNAME value of DBCTL. To inquire about the connection to DB2, use an EXITPROGRAM value of DFHD2EX1, or DSN2EXT1 (DSN2EXT1 is still recognised for compatibility with earlier releases) , with an ENTRYNAME of DSNCSQL, or DSNCCMD.

CVDA values are:

CONNECTED

The task-related user exit is connected to its external resource manager subsystem, and API requests can be issued.

NOTAPPLIC

The exit is not a task-related user exit.

NOTCONNECTED

The task-related user exit is not connected to its external resource manager subsystem, and therefore API requests cannot be issued.

UNKNOWN

The task-related user exit has been enabled and started, but not enabled for SPI requests. UNKNOWN can also be returned if CICS is unable to call the task related user exit. In both of these cases, CICS cannot tell whether it is connected to its external resource manager.

UNKNOWN is returned for all subsequent calls for the remaining lifetime of the task. A new task is able to call the task-related user exit and get the required information.

If the task-related user exit is not enabled, the INQUIRE command returns PGMIDERR. This also indicates that CICS is not connected to the resource manager.

Note: To determine whether DB2 or DBCTL is available, use CONNECTST rather than STARTSTATUS, because the task-related user exit can be started without having succeeded in making its database manager available to CICS.

ENTRY(*ptr-ref*)

returns a fullword binary field indicating the entry address of the user exit.

ENTRYNAME(*data-area*)

specifies the 8-character name of the exit about which you are inquiring. If you omit ENTRYNAME, CICS assumes that the name of the exit is the same as the name of the load module specified in the EXITPROGRAM option.

Consequently, you must specify the same values for ENTRYNAME and EXITPROGRAM as were specified in the ENTRYNAME and PROGRAM options on the ENABLE command that created the exit. (EXITPROGRAM in this command corresponds to PROGRAM in an ENABLE command.)

EXIT(*data-value*) (global user exits only)

specifies the 8-character identifier of an exit point with which the exit about which you are inquiring is associated. You must specify an exit point when you inquire about a global user exit. Exit points do not apply to task-related user exits, however, and you must not specify this option when you inquire about such an exit.

EXITPROGRAM(*data-value*)

specifies the 8-character name of the load module associated with the exit about which you want information. This is the value that was specified in the PROGRAM option of the ENABLE command that defined the exit.

FORMATEDFST(*cvda*) (task-related user exits only)

returns a CVDA value indicating that the FORMATEDF option is enabled for the exit. FORMATEDF causes extra invocations of the exit for tasks executed under EDF, to format output screens and interpret input, and applies only to task-related user exits. CVDA values are:

FORMATEDF

FORMATEDF is turned on.

NOFORMATEDF

FORMATEDF processing is turned off.

NOTAPPLIC

This is a global user exit.

GAENTRYNAME (*data-area*)

returns the 8-character name of the user exit that owns the global work area used by the exit about which you are inquiring.

This value is returned only when the exit uses a global work area owned by another exit. Blanks are returned if it has allocated its own work area.

GALENGTH (*data-area*)

returns a halfword binary field indicating the length of the global work area for the exit.

Note: If a GALENGTH greater than 32767 has been defined (see GALENGTH for ENABLE PROGRAM for details), the response to this command reflects that higher value as follows:

- If you issued the INQUIRE EXITPROGRAM command at your terminal, the response shows a negative value for GALENGTH.
- If you issued the INQUIRE EXITPROGRAM command from a program, the high order bit of the response for GALENGTH is set. You must allow for this possibility when deciding what operation to next perform on the returned value.

GAUSECOUNT (*data-area*)

returns a halfword binary field indicating the total number of global or task-related user exits that are using the global work area owned by this exit. This count includes the owning exit program. A zero is returned if the exit is not the owner.

INDOUBTST (*cvda*)

returns a CVDA value indicating whether the task-related user exit is enabled with the INDOUBTWAIT keyword. CVDA values are:

NOTAPPLIC

The exit being inquired upon is a global user exit.

NOWAIT

The exit is not enabled with the INDOUBTWAIT keyword.

WAIT The exit is enabled with the INDOUBTWAIT keyword.

NUMEXITS (*data-area*) (**global user exits only**)

returns a halfword binary field indicating the number of global user exit points at which the exit is enabled. A zero is returned if this is a task-related user exit.

PURGEABLEST (*cvda*) (**task-related user exits only**)

returns a CVDA value indicating whether the task-related user exit is enabled with the PURGEABLE keyword. CVDA values are:

NOTAPPLIC

The exit being inquired upon is a global user exit.

NOTPURGEABLE

Tasks are not purgeable from CICS waits within the task-related user exit.

PURGEABLE

Tasks are purgeable from CICS waits within the task-related user exit.

QUALIFIER (*data-area*)

returns, for a task-related user exit that is enabled for SPI calls, the 8-character qualifier returned by the exit.

For global user exits and task-related user exits that are not enabled for SPI calls, returns blanks.

SHUTDOWNST(*cvda*) (task-related user exits only)

returns a CVDA value indicating whether the SHUTDOWN option is enabled for the exit. SHUTDOWN causes invocation during CICS shutdown, and applies only to task-related user exits. CVDA values are:

NOSHUTDOWN

The exit is not invoked when a CICS shutdown occurs.

NOTAPPLIC

This is a global user exit.

SHUTDOWN

The exit is invoked when a CICS shutdown occurs.

SPIST(*cvda*)

returns a CVDA value indicating whether the task-related user exit is enabled for SPI calls. CVDA values are:

NOSPI

The exit is not enabled for SPI.

NOTAPPLIC

The exit being inquired upon is a global user exit. This occurs only when the INQUIRE command is explicitly for a global user exit. For example:

```
INQUIRE EXITPROGRAM(abcd) exit(XFCREQ)
```

If you omit EXIT(XFCREQ), you are inquiring about a task-related user exit. Because all global user exits are, by default, task-related user exits as well, NOSPI is returned.

SPI The exit is enabled for SPI.

STARTSTATUS(*cvda*)

returns a CVDA value identifying whether the exit is available for execution. CVDA values are:

STARTED

The exit program is available for execution; that is, the START option on an EXEC CICS ENABLE command is still in force.

STOPPED

The exit program is not available for execution; that is, the START option has not been issued, or has been revoked by the STOP option on an EXEC CICS DISABLE command.

TALENGTH(*data-area*) (task-related user exits only)

returns a halfword binary field indicating the length of the local (task-related) work area for the exit. Local work areas apply only to task-related user exits. A zero is returned if this is a global user exit.

TASKSTARTST(*cvda*) (task-related user exits only)

returns a CVDA value indicating whether the TASKSTART option is enabled for the exit. TASKSTART causes CICS to invoke the exit at the start and end of every task; it applies only to task-related user exits. CVDA values are:

NOTAPPLIC

This is a global user exit.

NOTASKSTART

The exit is not set for invocation at the start and end of every task.

TASKSTART

The exit is set for invocation at the start and end of every task.

USECOUNT (*data-area*)

returns the number of times the exit program has been invoked.

Note: The value returned is the total number of times this exit program has been invoked at all the global user exit points and task related user exit invocation points that the exit program has been enabled and started at.

Conditions**END**

RESP2 values:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

INVREQ

RESP2 values:

- 3 The exit point identified by EXIT does not exist.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

PGMIDERR

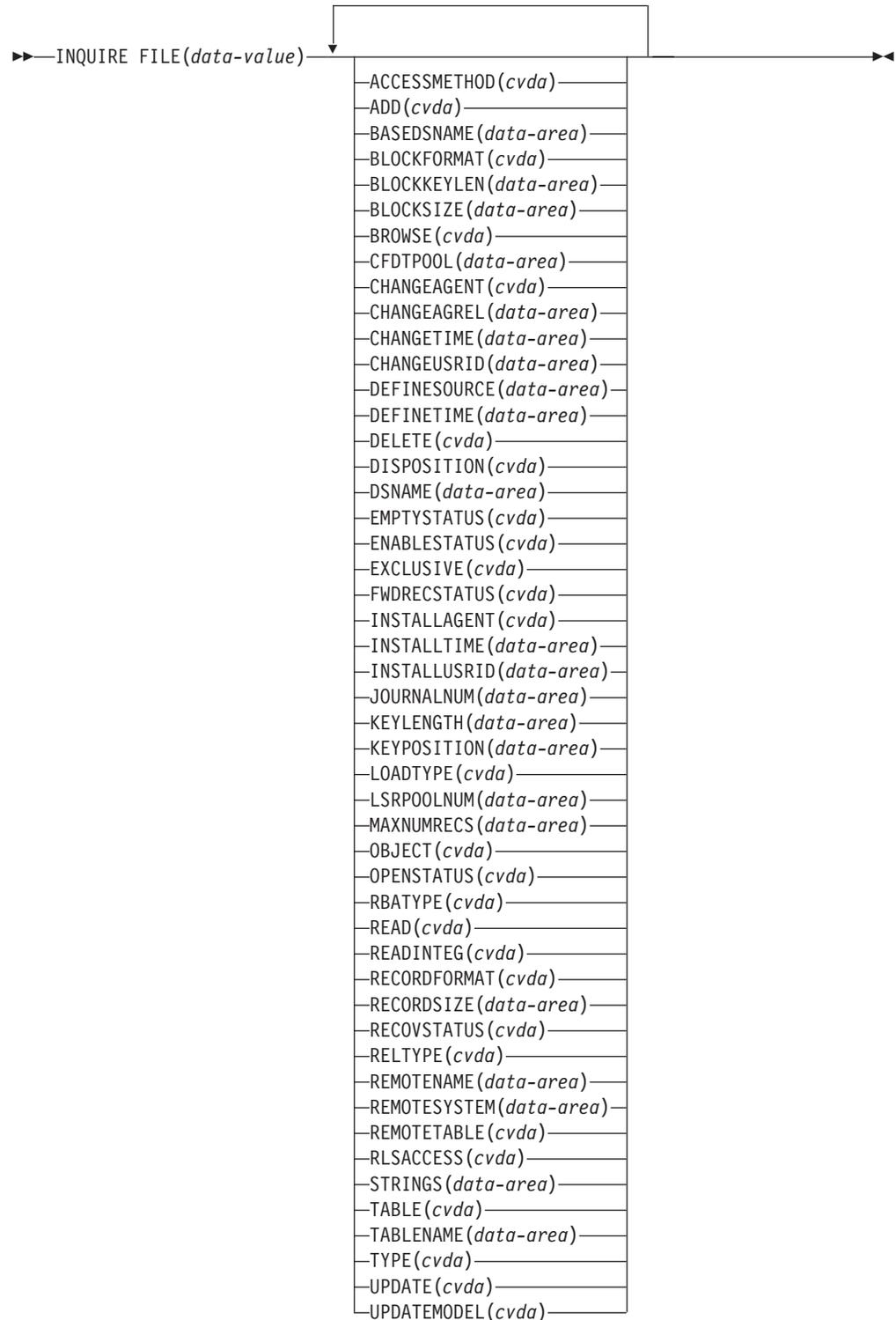
RESP2 values:

- 1 The exit identified by EXITPROGRAM and ENTRYNAME is not enabled, or the EXIT parameter is missing on an inquiry on a global user exit, or is present on a task-related user exit.

INQUIRE FILE

Retrieve information about a file.

INQUIRE FILE



Conditions: END, FILENOTFOUND, ILLOGIC, NOTAUTH

For more information about the use of CVDAs, see CICS-value data areas (CVDAs) in CICS System Programming Reference.

This command is threadsafe.

Description

The INQUIRE FILE command returns information about a FILE resource definition.

When the file is associated with a VSAM or BDAM object, INQUIRE FILE also returns information about the associated object.

- For VSAM, the object can be a base cluster (a KSDS, ESDS, or RRDS), an alternate index, or a path to a base cluster through an alternate index.
- For BDAM, the object is a single MVS BDAM data set.

(You cannot use the INQUIRE FILE command to get information about DL/I data sets or data sets associated with other CICS resources or functions. However, see the INQUIRE DUMPDS, JOURNALNAME, and TDQUEUE commands if you need information about dump data sets, journals, or TD queues.

The values returned depend on these criteria:

- Whether the file is open or closed and, if it is closed, whether it has been open during the current execution of CICS.

If the file is not open, you get default or null values, or values describing the most recent object associated with the file, as noted in the option descriptions that follow.

- Whether the file is local (defined on the same CICS system as the task making the inquiry) or remote (defined on another CICS system).

Less information is available for remote files; therefore, defaults or nulls are returned for some options.

For further information about null values, see Null values in CICS System Programming Reference.

- If a file is empty (in VSAM load mode), after the first write or mass insert has completed the file is closed and left enabled. It remains so until the next access (write or read) when it is implicitly opened.

If an INQUIRE is issued against the file before this next access occurs, the file shows CLOSED,ENABLED. This state can be temporary for a file that has just completed load mode.

Some options for the INQUIRE FILE command are specific to one or another of the file objects supported by CICS, such as VSAM or BDAM data sets, and data tables. You can specify many of these parameters even when the file refers to a different object from that to which the parameters apply. In this way it is easier to switch file definitions between different objects; for example, between non-RLS and RLS access, or between a user-maintained data table and a coupling facility data table. When a parameter is specified for an object to which the file does not currently refer, it is ignored.

Browsing

You can browse through all the files installed in your system by using the browse options on INQUIRE FILE commands: START, NEXT, and END. See Browsing resource definitions in CICS System Programming Reference for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

ACCESSMETHOD(*cvda*)

Returns a CVDA value identifying the access method for this file. CVDA values are as follows:

BDAM

The access method is BDAM.

REMOTE

The file is defined as remote, and therefore the access method is not known to the local CICS system.

VSAM

The access method is VSAM. Access to a data table (except while it is being loaded or, for a CICS-maintained data table, when the source data set is being updated or searched for a record that is not in the table), is through CICS data table services. Because this access is still based on VSAM keys, CICS returns VSAM as the access method for any kind of data table.

ADD(*cvda*)

Returns a CVDA value identifying whether new records can be added to the file. CVDA values are as follows:

ADDABLE

New records can be added to the file.

NOTADDABLE

New records cannot be added to the file.

NOTAPPLIC

The value does not apply to this type of file. For example, it does not apply to a remote file.

BASEDSNAME(*data-area*) (VSAM only)

Returns the 44-character name of the base cluster associated with a VSAM path, if the object associated with the file is a path. If the object is other than a path, this option returns the same value as the DSNAME option.

BASEDSNAME is blank if the file has not been opened since the last initial or cold start of this CICS. If the file has been opened at least once since the last initial or cold start, CICS returns the 44-character name, even though the file might not be open at the time the command is issued. CICS can return the name because the name is preserved in the CICS catalog and recovered on a restart.

If the object is a coupling facility data table loaded from a source data set, the 44-character name returned on BASEDSNAME is the same as that returned on DSNNAME. BASEDSNAME is blank for a coupling facility data table that is not associated with a source data set.

The translator still accepts BASENAME for this option, but uses BASEDSNAME in new code.

BLOCKFORMAT(*cvda*) (BDAM only)

Returns a CVDA value identifying whether records on the file are blocked or unblocked. CVDA values are as follows:

BLOCKED

The records on the file are blocked, or this file is a VSAM file.

UNBLOCKED

The records on the file are unblocked.

NOTAPPLIC

The value does not apply to this type of file. For example, it does not apply to a remote file.

BLOCKKEYLEN(*data-area*) (BDAM only)

Returns a fullword binary field indicating the physical block key length for the file.

BLOCKSIZE(*data-area*) (BDAM only)

Returns a fullword binary field indicating the length in bytes of a block. If the blocks are of variable length or are undefined, the value returned is the maximum.

BROWSE(*cvda*)

Returns a CVDA value identifying whether you can browse the file. CVDA values are as follows:

BROWSABLE

You can browse the file.

NOTBROWSABLE

You cannot browse the file.

NOTAPPLIC

The value does not apply to this type of file. For example, it does not apply to a remote file.

CFDTPOOL(*data-area*) (CFDT only)

Returns the 8-character name of the coupling facility data table pool in which the coupling facility data table resides. CICS returns blanks if the file does not refer to a coupling facility data table and no pool name has been specified.

CHANGEAGENT(*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

SYSTEM

The resource definition was last changed by the CICS or CICSplex system.

TABLE

The resource definition was last changed by a table definition.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see FORMATTIME in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

DELETE (*cvda*) (**VSAM only**)

Returns a CVDA value identifying whether you can delete records from the file. CVDA values are as follows:

DELETABLE

You can delete records from the file.

NOTDELETABLE

You cannot delete records from the file.

NOTAPPLIC

The value does not apply to this type of file. For example, it does not apply to a remote file.

DISPOSITION (*cvda*)

Returns a CVDA value indicating the value of the DISPOSITION option for the file, from the DISPOSITION option in the FILE definition or the JCL DD statement to which it points. CVDA values are as follows:

OLD Disposition is OLD.

SHARE

Disposition is SHARE.

NOTAPPLIC

The value does not apply to this type of file. For example, it does not apply to a remote file.

DSNAME (*data-area*)

Returns the 44-character name of the BDAM data set or VSAM object associated with the FILE definition.

If the file has not been opened since the last initial or cold start, the name is taken from the file resource definition. CICS returns blanks if the data set name is not defined on the file definition.

For a coupling facility data table loaded from a data set, CICS returns the 44-character source data set name. For a coupling facility data table that is not loaded from a data set, CICS returns blanks.

EMPTYSTATUS(*cvda*) (VSAM only)

Returns a CVDA value indicating whether EMPTYREQ has been set for the file. EMPTYREQ causes the object associated with this file to be set to empty, if eligible, when the file is opened. VSAM data sets defined as reusable, and defined to be used in non-RLS mode, are the only ones that you can make empty in this way; EMPTYREQ has no effect on other objects. CVDA values are as follows:

EMPTYREQ

The data set must be made empty.

NOEMPTYREQ

The data set must not be made empty.

NOTAPPLIC

The value does not apply to this type of file. For example, it does not apply to a remote file.

ENABLESTATUS(*cvda*)

Returns a CVDA value identifying whether application programs can access the file. CVDA values are as follows:

DISABLED

The file is unavailable for access by application programs because it has been explicitly disabled. It must be explicitly enabled by a SET FILE ENABLED command or its CEMT equivalent before it can be accessed by application programs.

DISABLING

A request to disable the file has been received, but tasks are running that had previously accessed the file. These tasks are allowed to complete their use of the file, but new tasks are not allowed access.

ENABLED

The file is available for access by application programs.

UNENABLED

The file is unavailable for access by application programs because it is closed. It must be explicitly enabled by a SET FILE OPEN command or its CEMT equivalent before it can be accessed by application programs.

UNENABLING

A request to close the file has been received, but tasks are running that previously accessed the file. These tasks are allowed to complete their use of the file, but new tasks are not allowed access.

NOTAPPLIC

The value does not apply to this type of file. For example, it does not apply to a remote file.

EXCLUSIVE(*cvda*) (BDAM only)

Returns a CVDA value identifying whether records on this file are to be placed under exclusive control when a read for update is issued. CVDA values are as follows:

EXCTL

A record on this file is placed under exclusive control of the reading task when it is read for update.

NOEXCTL

A record on this file is not placed under exclusive control when it is read for update.

NOTAPPLIC

The value does not apply to this type of file. For example, it does not apply to a remote file.

FILE(*data-value*)

Specifies the 8-character name of the file about which you are inquiring.

FWDRECSTATUS(*cvda*) (**VSAM only**)

Returns a CVDA value identifying whether the file is forward-recoverable.

The value CICS returns for FWDRECSTATUS depends on whether the file has been opened since the last initial or cold start:

- If the file has not been opened since the last initial or cold start, CICS returns the value from the file definition.
- If the file has been opened at least once since the last initial or cold start, CICS returns the value that was used when the file was last opened. This value can be different from the value on the file definition because, for example, the file definition can be overridden by a value from the ICF catalog.

CVDA values are as follows:

FWDRECOVERABLE

The file is forward-recoverable. The RECOVERY option of the FILE definition specifies that updates to the file are to be recorded, to make forward recovery of the file possible. The forward-recovery log can be found using INQUIRE DSNAME.

NOTFWDRCVBLE

The file is not forward-recoverable. CICS returns NOTFWDRCVBLE for a coupling facility data table and a user-maintained data table.

NOTAPPLIC

The value does not apply to this type of file. For example, it does not apply to a remote file.

INSTALLAGENT(*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

SYSTEM

The resource was installed by the CICS or CICSplex SM system.

TABLE

The resource was installed by using a table definition.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

JOURNALNUM(*data-area*)

Returns a halfword binary field indicating the number of the journal to which CICS writes the information required for autojournaling. The value returned in JOURNALNUM is the number specified by the JOURNAL parameter in the file resource definition.

Journal numbers are between 1 and 99 and correspond to journal names DFHJ01 through DFHJ99. A value of 0 means that JOURNAL(NO) is specified and CICS does not perform autojournaling for the file.

JOURNALNUM is ignored for user-maintained and coupling facility data tables. Requests made to these tables are not autojournalled.

KEYLENGTH(*data-area*)

Returns a fullword binary field indicating the length of the record key for a file associated with a VSAM KSDS or a file associated with a coupling facility data table. If the file is associated with a BDAM data set, the value is the length of the logical key used for deblocking.

Note:

1. If the file is closed and the key length is not defined in the file definition, the value returned is 0 (zero).
2. If the file is closed and a key length is defined on the file definition, CICS returns the value from the file definition.
3. If the file is open, most files get their key length from the associated data set, in which case CICS returns the value from the data set. However, files that refer to coupling facility data tables defined with LOAD(NO) must get their keylength from the file definition, in which case CICS returns the value from the file definitions for such files. This value must also match that of the coupling facility data table if it has already been created.

KEYPOSITION(*data-area*)

Returns a fullword binary field indicating the starting position of the key field in each record relative to the beginning of the record. The start is made at position 0. If there is no key, or if the file is not open, CICS returns a value of zero for the key position.

For a coupling facility data table associated with a source data set, where the file is open, the key position is obtained from the source data set. If the coupling facility data table is not associated with a source data set, CICS returns zero.

LOADTYPE(*cvda*) (**VSAM only**)

Returns a CVDA value indicating the load type for a coupling facility data table. CVDA values are as follows:

LOAD

The coupling facility data table is preloaded from a source data set.

NOLOAD

The coupling facility data table is not preloaded from a source data set.

NOTAPPLIC

The file is not defined as a coupling facility data table, and no value is defined in the file resource definition.

CICS returns LOAD or NOLOAD if the file is not defined as a coupling facility data table, but one of these options is specified on the LOAD attribute of the file resource definition. In this case, the LOADTYPE CVDA indicates the load type that applies if the file definition is altered to specify TABLE(CF).

LOADTYPE has no significance for a CICS-maintained or user-maintained shared data table. A shared data table is always loaded from a source data set when the first file to reference the table is opened.

LSRPOOLID(*data-area*) (VSAM only)

This attribute is obsolete, but is supported to provide compatibility with earlier releases of CICS.

A value specified for LSRPOOLID is transferred to the new option LSRPOOLNUM.

LSRPOOLNUM(*data-area*) (VSAM only)

Returns a fullword binary field indicating the number of the VSAM LSR pool associated with this file, in the range 1 through 255. If the file does not share buffers, the LSRPOOLNUM value is 0.

MAXNUMRECS(*data-area*) (data tables only)

Returns a fullword binary field indicating the maximum number of records that the data table for this file can hold. The value returned by CICS is affected by the following factors:

- If the file resource definition specifies a MAXNUMRECS numeric value, even though the object is not a table (NOTTABLE CVDA is returned on the TABLE option), CICS returns the specified value.
- If the file resource definition is specified with MAXNUMRECS(NOLIMIT), meaning that the number of records is unlimited, CICS returns a value of zero. Internally, CICS holds NOLIMIT as the maximum positive fullword value (+2147483647 or X'7FFFFFFF').
- If the file is remote, CICS returns a value of minus 1 (-1).
- If the object is a coupling facility data table, the following points apply:
 - The maximum number of records can be altered by a coupling facility data table server command, leaving the file definition MAXNUMRECS value unchanged. CICS returns the value in the file definition until the file is opened, after which CICS returns the actual MAXNUMRECS value defined to the server.
 - If the value is changed again by a coupling facility data table server command, CICS obtains and returns the new value only after the file is next opened or inquired on. Until then, CICS continues to return the old value.
 - You can use the server DISPLAY TABLE console command to obtain the current value for a coupling facility data table.

OBJECT(*cvda*) (VSAM only)

Returns a CVDA value indicating whether the file is associated with a data set (a VSAM KSDS, ESDS, or RRDS, or an alternate index used directly) or a VSAM path that links an alternate index to its base cluster. CVDA values are as follows:

BASE The file is associated with a data set that is a VSAM base. CICS also

returns BASE for data tables. (Data table access provides primary key access only, not access through a path.)

PATH The file is associated with a path. You get a value of PATH only if the file defines a path to a VSAM base data set through an alternate index. If the file definition allows direct access to an alternate index, or if the path is used merely as an alias to a base data set, you get a value of BASE. Also, if the file has not been opened since the last initial or cold start, CICS returns a default value of BASE.

NOTAPPLIC

The value does not apply to this type of file. For example, it does not apply to a remote file.

If the file is a data table, the OBJECT option refers to its source data set.

OPENSTATUS (*cvda*)

Returns a CVDA value identifying whether the file is open, closed, or in a transitional state. The OPENSTATUS value affects the ability of application tasks to access the file, but only indirectly; see the ENABLESTATUS option description for the rules. CVDA values are as follows:

CLOSED

The file is closed.

CLOSEREQUEST

The file is open and in use by one or more application tasks. An EXEC CICS SET FILE CLOSED or a CEMT SET FILE CLOSED request has been received, but closing is not complete (the ENABLESTATUS of the file is DISABLING).

NOTAPPLIC

The OPENSTATUS value does not apply to this type of file. For example, it does not apply to a remote file.

OPEN The file is open.

RBATYPE (*cvda*)

Returns a CVDA value identifying whether, for VSAM files, the data set uses extended addressing. CVDA values are as follows:

EXTENDED

This VSAM data set uses extended relative byte addressing and therefore can hold more than 4 GB of data.

NOTAPPLIC

One of the following is true:

- The data set is BDAM.
- The file is remote.
- The file is not open.

NOTEXTENDED

This VSAM data set does not use extended relative byte addressing and therefore cannot hold more than 4 GB of data.

READ (*cvda*)

Returns a CVDA value identifying whether you can read records from the file. CVDA values are as follows:

NOTREADABLE

You cannot read records from the file.

READABLE

You can read records from the file.

NOTAPPLIC

The value does not apply to this type of file. For example, it does not apply to a remote file.

READINTEG(*cvda*)

Returns a CVDA value indicating the default level of read integrity that is active for the file if a read integrity option is not explicitly coded on a file read request command. CVDA values are as follows:

CONSISTENT

Read requests for this file are subject to consistent read integrity, unless otherwise specified on the read request.

NOTAPPLIC

Read integrity is not applicable for this file for one of the following reasons:

- The file is a VSAM file accessed in non-RLS mode.
- The file is a remote file.
- The file refers to a BDAM data set.
- The file refers to a coupling facility data table.

If you switch a file from RLS to non-RLS mode, the read integrity option specified for RLS mode is preserved. In this case, CICS returns NOTAPPLIC. If you switch the file back to RLS mode, CICS returns the saved read integrity in response to an INQUIRE FILE command.

REPEATABLE

Read requests for this file are subject to repeatable read integrity, unless otherwise specified on the read request.

UNCOMMITTED

No read integrity is specified for this file.

RECORDFORMAT(*cvda*)

Returns a CVDA value identifying the format of the records on the file. CVDA values are as follows:

FIXED

The records are of fixed length.

UNDEFINED

The format of records on the file is undefined. The UNDEFINED value is possible for BDAM data sets only.

VARIABLE

The records are of variable length. If the file is associated with a user-maintained data table, the record format is always variable length, even if the source data set contains fixed-length records.

NOTAPPLIC

The value does not apply to this type of file. For example, it does not apply to a remote file.

RECORDSIZE(*data-area*)

Returns a fullword binary field indicating the actual size of fixed-length records, or the maximum size of variable-length records.

If the file is not open, CICS returns the value specified in the installed file definition.

If the file is open, most files get their record size from the associated data set, in which case CICS returns the value from the data set. However, files that refer to coupling facility data tables defined with LOAD(NO) must get their record size from the file definition, in which case CICS returns the value from the file definitions for such files. This value must also match that of the coupling facility data table if it has already been created.

RECOVSTATUS(*cvda*)

Returns a CVDA value identifying whether the file is recoverable.

The value CICS returns for RECOVSTATUS depends on whether the file has been opened since the last initial or cold start of the CICS region:

- If the file has not been opened since the last initial or cold start, CICS returns the value from the file definition.
- If the file has been opened at least once since the last initial or cold start, CICS returns the value that was used when the file was last opened. This value can be different from the value on the file definition because, for example, the file definition might be overridden by a value from the ICF catalog. Any value from the ICF catalog is ignored for a user-maintained or CICS-maintained data table.

CVDA values are as follows:

NOTRECOVERABLE

The file is not recoverable.

RECOVERABLE

The file is recoverable.

NOTAPPLIC

The value does not apply to this type of file. For example, it does not apply to a remote file.

RELTYPE(*cvda*) (**BDAM only**)

Returns a CVDA value indicating whether relative or absolute addressing is used to access the file and, if relative, the type of relative addressing. CVDA values are as follows:

BLK Relative block addressing is being used.

DEC The zoned decimal format is being used.

HEX The hexadecimal relative track and record format is being used.

NOTAPPLIC

Absolute (MBBCCCHR) addressing is being used or the file is a VSAM file.

REMOTENAME(*data-area*)

Returns the 8-character name by which the file is known in the CICS region named in the REMOTESYSTEM option of its FILE definition. Blanks are returned if the file is not remote.

REMOTESYSTEM(*data-area*)

Returns a 4-character name of the CICS region in which the file is defined, from the REMOTESYSTEM value in the FILE definition. Blanks are returned if the file is not remote.

REMOTETABLE(*cvda*) (**VSAM only**)

Returns a CVDA value indicating whether the file represents an open remote data table. Only one CVDA value applies as follows:

REMTABLE

The file represents an open remote data table.

RLSACCESS(*cvda*)

Returns a CVDA value indicating whether the file is defined to be opened in RLS mode. CVDA values are as follows:

NOTAPPLIC

The file is not eligible to be accessed in RLS mode if it is a remote file, or if it refers to a BDAM data set.

NOTRLS

The file refers to a data set defined to be accessed in non-RLS mode.

RLS The file refers to a data set defined to be accessed in RLS mode.

STRINGS(*data-area*) (**VSAM only**)

Returns a fullword binary field indicating the number of strings (concurrent operations) specified for the file in its FILE definition.

TABLE(*cvda*) (**VSAM and CFDT only**)

Returns a CVDA value indicating whether the file represents a data table. CVDA values are as follows:

CFTABLE

The file represents a coupling facility data table.

CICSTABLE

The file represents a CICS-maintained data table.

NOTTABLE

The file does not represent a data table.

USERTABLE

The file represents a user-maintained data table.

NOTAPPLIC

The value does not apply to this type of file. For example, it does not apply to a remote file.

TABlename(*data-area*) (**CFDT only**)

Returns the 8-character table name specified for the coupling facility data table on the file resource definition, if one is specified. Returns the file name if the table name is omitted from the file resource definition.

CICS returns blanks if the file does not refer to a coupling facility data table.

TYPE(*cvda*)

Returns a CVDA value identifying the type of data set that corresponds to this file. The data set must be open to return the type of data set. CVDA values are as follows:

ESDS The data set is an entry-sequenced data set.

KEYED

The data set is addressed by physical keys.

KSDS The data set is a key-sequenced data set or the file refers to a data table.

NOTKEYED

The data set is not addressed by physical keys.

RRDS The data set is a relative record data set.

VRRDS

The data set is a variable-length relative record data set.

NOTAPPLIC

The data set is not open.

UPDATE(*cvda*)

Returns a CVDA value identifying whether the file is updatable. CVDA values are as follows:

NOTUPDATABLE

You cannot update records.

UPDATABLE

You can update records.

NOTAPPLIC

The value does not apply to this type of file. For example, it does not apply to a remote file.

UPDATEMODEL(*cvda*) (CFDT only)

Returns a CVDA value indicating the update model specified for the coupling facility data table in the installed file definition. CVDA values are as follows:

CONTENTION

The coupling facility data table is updated using the contention model.

LOCKING

The coupling facility data table is updated using the locking model.

NOTAPPLIC

The file does not refer to a coupling facility data table and UPDATEMODEL on the file resource definition does not specify a value.

You can define a file that specifies LOCKING or CONTENTION on the UPDATEMODEL attribute when the file does not refer to a coupling facility data table. In this case, CICS returns the specified UPDATEMODEL value on the INQUIRE FILE command, and not NOTAPPLIC. If you redefine the command to refer to a coupling facility data table, the specified UPDATEMODEL takes effect.

For information about the contention and locking models, see FILE attributes in the Resource Definition Guide.

Conditions**END**

RESP2 values:

2 There are no more resource definitions of this type.

FILENOTFOUND

RESP2 values:

1 The file cannot be found.

ILLOGIC

RESP2 values:

1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

INQUIRE HOST

Retrieve information about virtual hosts in the local system.

INQUIRE HOST



Conditions: END, ILLOGIC, INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

You can also browse through all the virtual hosts that exist in the region, using the browse options (START, NEXT, and END) on INQUIRE HOST commands. See *Browsing resource definitions* for general information about browsing, including syntax, exception conditions, and examples.

Options

HOST(*data-value*)

specifies the name of a virtual host. The name of each virtual host is taken from the host name specified in the URIMAP definitions that make up the virtual host. For example, if your CICS region contained URIMAP definitions that specified a host name of `www.example.com`, CICS would create a virtual host with the name `www.example.com`. A host name in a URIMAP definition can be up to 120 characters.

ENABLESTATUS(*cvda*)

returns a CVDA value indicating the status of this virtual host. CVDA values are:

ENABLED

The virtual host is enabled.

DISABLED

The virtual host is disabled. The URIMAP definitions that make up the virtual host cannot be accessed by applications.

TCPIPSERVICE(*data-area*)

returns the 1- to 8-character name of the TCPIPSERVICE definition that specifies the inbound port to which this virtual host relates. If this definition is not given, the virtual host relates to all TCPIPSERVICE definitions.

Conditions

END

RESP2 values are:

- 2 There are no more virtual hosts.

ILLOGIC

RESP2 values are:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

INVREQ

RESP2 values are:

- 10 The specified host name contains disallowed characters, or is blank.

NOTAUTH

RESP2 values are:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

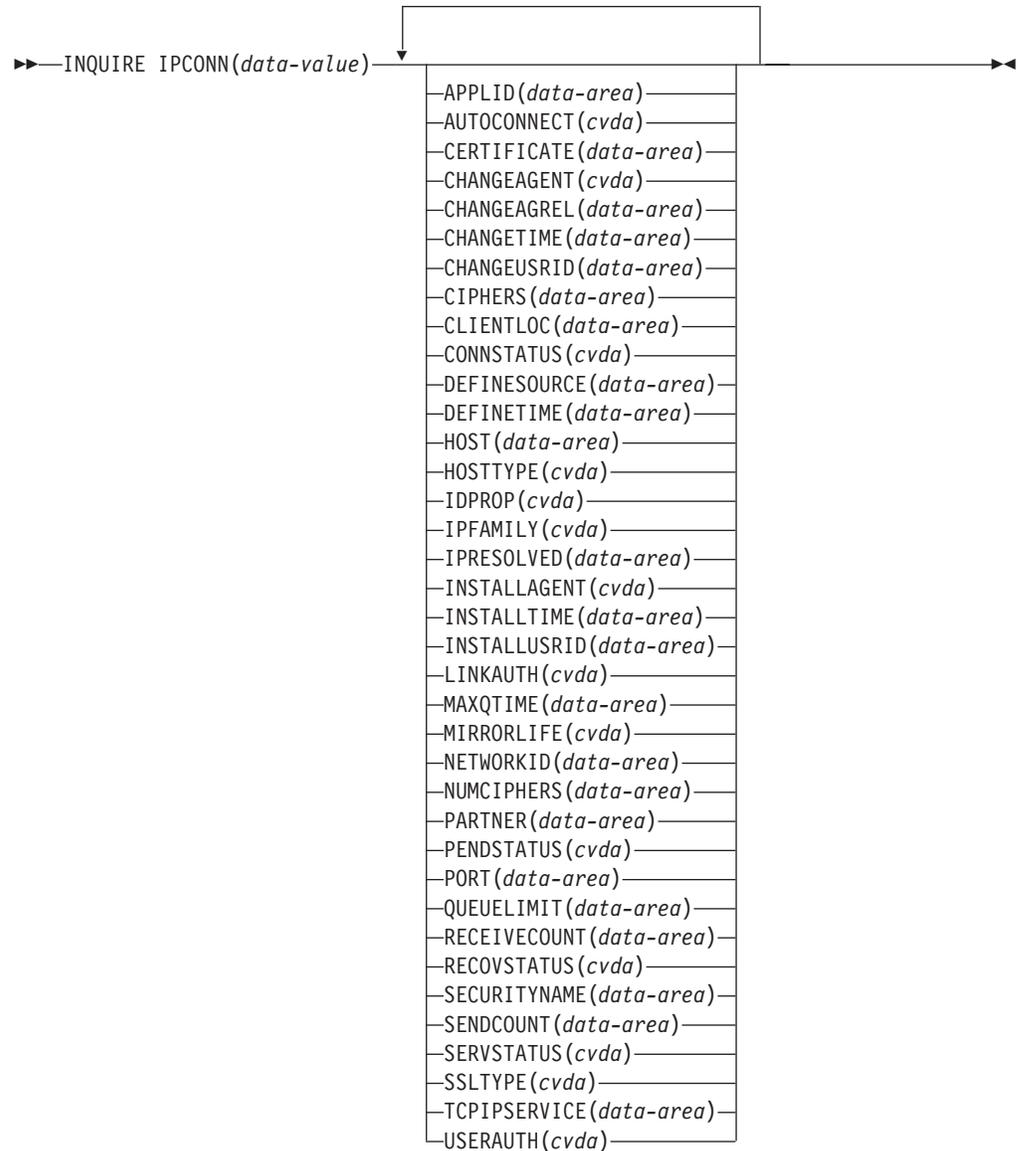
RESP2 values are:

- 5 The virtual host cannot be found.

INQUIRE IPCONN

Retrieve information about an IPIC connection.

INQUIRE IPCONN



Conditions: END, ILLOGIC, NOTAUTH, SYSIDERR

For more information about the use of CVDAs, see CICS-value data areas (CVDA) in CICS System Programming Reference.

This command is threadsafe.

Description

The INQUIRE IPCONN command retrieves information about an IPIC connection. An IPCONN resource is a Transmission Control Protocol/Internet Protocol (TCP/IP) communication link from your local CICS region to another CICS region or another system.

Note:

- See also INQUIRE CONNECTION in CICS System Programming Reference. The INQUIRE CONNECTION command returns information about MRO and ISC over SNA connections.

For information about the different kinds of intercommunication connections, see the *CICS Intercommunication Guide*.

- The *outbound* attributes of the IPIC connection are specified by an IPCONN definition. The *inbound* attributes of the connection are specified by the TCPIP SERVICE definition named on the TCPIP SERVICE option of the IPCONN definition.

Browsing

You can also browse through all of the IPCONN definitions installed in your system by using the browse options (START, NEXT, and END) on INQUIRE IPCONN commands. See Browsing resource definitions in CICS System Programming Reference for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

APPLID(*data-area*)

Returns the 8-character name by which the remote system is known to the network, taken from the APPLID option of the IPCONN definition. This name is the application identifier (*applid*) of the remote system, as specified on the APPLID option of its system initialization parameter. For XRF systems, it is the generic APPLID.

AUTOCONNECT(*cvda*)

Returns a CVDA value identifying which AUTOCONNECT option has been specified in the IPCONN definition. CVDA values are as follows:

AUTOCONN

AUTOCONNECT(YES) has been specified on the IPCONN definition.

NONAUTOCONN

AUTOCONNECT(NO) has not been specified for the IPCONN definition.

CERTIFICATE(*data-area*)

Returns a 32-character area containing the label of the certificate, in the key ring, that is used as a client certificate in the SSL handshake for outbound IPCONN connections. If the label is blank, the certificate nominated as the default for the key ring is used.

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

AUTOINSTALL

The resource was autoinstalled.

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

CIPHERS (*data-area*)

Returns a 56-character area containing the list of cipher suites that is used to negotiate with clients during the SSL handshake. The list is set by the **ENCRYPTION** system initialization parameter, but you can edit the list to remove or change the order of cipher suites. See the *CICS RACF Security Guide*.

CLIENTLOC (*data-area*)

Returns a 32-character area that represents an evaluation of the **SO_CLUSTERCONNTYPE** options returned by z/OS Communications Server, for all the sockets used by the IPIC connection. See the z/OS 1.11 Communications Server IP Sockets Application Programming Interface Guide, http://publib.boulder.ibm.com/infocenter/zos/v1r9/index.jsp?searchWord=SO_CLUSTERCONNTYPE&tab=search , in the z/OS 1.11 information center for a description of **SO_CLUSTERCONNTYPE** and an explanation of the bit settings. Multiple sockets might provide the IPIC connection with a number of different paths to the partner system. Each character in **CLIENTLOC** is displayed as either zero or one. **CLIENTLOC** represents the most diverse route between the CICS region and its partner system.

CONNSTATUS (*cvda*)

Returns a CVDA value identifying the state of the IPIC connection between CICS and the remote system. CVDA values are as follows:

ACQUIRED

The IPIC connection is acquired. The criterion for **ACQUIRED** is that the capabilities exchange is complete. The capabilities exchange is the

way that two connected CICS regions discover the levels of service that they can collectively support; for example, the sync point level, and security protocols such as SSL.

FREEING

The IPIC connection is being released.

OBTAINING

The IPIC connection is being acquired. The connection remains in the OBTAINING state until all the criteria for ACQUIRED have been met.

RELEASED

The IPIC connection is RELEASED. Although it might also be in INSERVICE status, it is not usable.

The RELEASED status can be caused by any one of the following conditions:

- The remote system has not yet initialized.
- No IPCONN definition exists on the remote system and autoinstall was not active or not successful.
- The IPCONN definition on the remote system has been set out of service.
- AUTOCONNECT(NO) has been specified on the IPCONN definition.
- The IPIC connection had been acquired but has since been released by an explicit operator command.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

HOST (*data-area*)

Returns the 116-character host name of the remote system or its IPv4 or IPv6 address. The HOST option can be a character host name, an IPv4 address, or an IPv6 address. HOST is specified in the resource definition. HOST displays all IPv4 addresses as native IPv4 dotted decimal addresses, for example, 1.2.3.4, irrespective of which type of address format is used. You can specify IPv4 and IPv6 addresses in a number of acceptable formats. See IP addresses in the Internet Guide for more information on address formats.

HOSTTYPE (*cvda*)

Returns the address format of the HOST option. HOSTTYPE is set by the domain when the IPIC connection is installed. CVDA values are as follows:

HOSTNAME

The HOST option contains a character host name. The IP address that corresponds to the host name is looked up in the domain name server.

IPV4 The address is an IPv4 address.

IPV6 The address is an IPv6 address.

NOTAPPLIC

An incorrect host address was returned (HOST=0.0.0.0).

IDPROP(*cvda*)

Indicates whether the sender includes the distributed identity in requests over the IPIC connection. The IDPROP option is meaningful only if a connection extends outside a sysplex and is used primarily to prevent distributed identities being transmitted between enterprises. If the connection is between systems in the same sysplex, the value returned by this option is ignored, and the connection operates as if IDPROP(OPTIONAL) is specified.

CDVA values are as follows:

NOTALLOWED

A user ID associated with the sending transaction is sent for requests using this connection. NOTALLOWED is the default value.

OPTIONAL

A distributed identity is sent, if available. The user ID associated with the sending transaction is also sent.

REQUIRED

A distributed identity is required for requests using this connection. If REQUIRED is specified, the receiving system must support distributed identities. The user ID associated with the sending transaction is not sent.

INSTALLAGENT(*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

AUTOINSTALL

The resource was autoinstalled.

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

IPCONN(*data-value*)

Returns the 8-character identifier of the remote system or region about which you are inquiring; that is, the name assigned to its IPCONN definition.

IPFAMILY(*cvda*)

Returns the address format of the IPRESOLVED option. IPFAMILY is set only when the IPIC connection is acquired. CDVA values are as follows:

IPV4 The address is specified in IPv4 dotted decimal address format.

IPV6 The address is specified in IPv6 colon hexadecimal address format.

UNKNOWN

The IPRESOLVED option is not yet in use or the address cannot be resolved. UNKNOWN is the default when IPRESOLVED is 0.0.0.0.

IPRESOLVED(*data-area*)

Returns a 39-character field that specifies the IPv4 or IPv6 address of the HOST option. If the IPCONN resource has not yet been acquired or has been released, or the address cannot be resolved, a default value of 0.0.0.0 is returned. After the IPIC connection is acquired, IPRESOLVED displays the last resolved IP address that was used by the IPCONN resource. IPRESOLVED is reset to 0.0.0.0 when the resource is out of service and released. The content of IPRESOLVED is not recoverable after a warm or emergency restart.

LINKAUTH(*cvda*)

Returns a CVDA value that specifies how the user ID for link security is established in a CICS system with security initialized (SEC=YES).

CERTUSER

TCP/IP communication with the partner system must be configured for SSL and a certificate must be received from the partner system during SSL handshake.

The IPCONN must refer to a TCPIPSERVICE that is defined with SSL(CLIENTAUTH).

The received certificate must be defined to the external security manager so that it is associated with a user ID, which is used to establish link security.

SECUSER

Specifies that the user ID specified in SECURITYNAME is used to establish link security.

This value is the default.

MAXQTIME(*data-area*)

Returns a fullword binary value giving the maximum time, in seconds, for which allocate requests can be queued. The value is in the range 0 - 9999, or has the standard null value of -1 if MAXQTIME(NO) is specified on the IPCONN definition.

MIRRORLIFE(*cvda*)

Returns the minimum lifetime of the mirror task for function-shipped file control, transient data, and temporary storage requests received by this region. CVDA values are as follows:

REQUEST

The mirror task terminates as soon as possible. This is the default value.

TASK The mirror task remains available to the application that issues the remote request until the task of the application ends.

UOW The mirror transaction remains available to the application that issues the remote request until the next sync point is issued.

NETWORKID(*data-area*)

Returns the network ID of the remote system. The value returned is an 8-byte character string, which is the value of the NETWORKID option of the IPCONN definition. If NETWORKID is not specified on the IPCONN definition, the value returned is the z/OS Communications Server NETID or, for the z/OS Communications Server VTAM=NO systems, the value of the UOWNETQL system initialization parameter of this CICS; that is, the CICS on which the IPCONN definition is installed.

The NETWORKID is used with the APPLID to ensure unique naming for connecting systems.

NUMCIPHERS (*data-area*)

Returns a binary halfword data area that contains the number of cipher suites that are specified in the CIPHERS attribute.

PARTNER (*data-area*)

Returns a 64-character field indicating the product token of the partner system. The field is blank when the connection is not acquired or if the partner system does not indicate a product type when the connection is established. For example, the partner system is IBM_CICS_Transaction_Server/4.1.0(zOS) for a CICS TS 4.1 partner.

PENDSTATUS (*cvda*)

Shows whether this IPIC connection has any pending units of work. CDVA values are as follows:

NOTPENDING

No mismatch of lognames has occurred with the partner.

PENDING

Resynchronization work is outstanding for the connection but the partner system has performed an initial start, preventing completion of the resynchronization process. You can use the SET IPCONN NOTPENDING command to unilaterally commit or back out the units of work associated with the connection, according to their associated transaction definitions. You can also investigate the units of work individually and force them to commit or back out, in which case you must also complete the recovery activity by using a SET IPCONN NOTPENDING command to clear the PENDING condition.

If this IPIC connection is CICS-to-CICS, no new sync point work, that is, work involving sync level 2 protocols, can be transmitted across the connection until a SET IPCONN NOTPENDING command has been issued.

If you are not concerned by the loss of synchronization caused by the initial or cold start of the partner, you can cause the SET IPCONN NOTPENDING command to be issued automatically by specifying XLNACTION(FORCE) on the IPCONN definition.

For further information about pending units of work, see the *CICS Intercommunication Guide*.

PORT (*data-area*)

Returns a fullword binary value, in the range 1 through 65535, containing the port number to be used for outbound requests on this IPIC connection; that is, the number of the port on which the remote system listens.

If the IPIC connection is defined with PORT(NO), the value is -1.

QUEUELIMIT (*data-area*)

Returns a fullword binary value giving the maximum number of allocate requests that can be queued for this IPIC connection. The value is in the range 0 - 9999, or has the standard null value of -1 if QUEUELIMIT(NO) is specified on the IPCONN definition.

RECEIVECOUNT (*data-area*)

Returns a fullword binary value giving the number of RECEIVE sessions defined for this IPIC connection.

RECOVSTATUS (*cvda*)

Returns a CVDA value indicating whether resynchronization work is outstanding for the IPIC connection. The connection might never have been connected, have been quiesced and all resynchronization work completed, or disrupted without quiesce, in which case resynchronization might be necessary. CVDA values are as follows:

NORECOVDATA

Neither side has recovery information outstanding.

NRS CICS does not have recovery outstanding for the connection, but the partner might have.

RECOVDATA

Indoubt units of work are associated with the connection, or outstanding resynchronization tasks are awaiting FORGET on the connection. Resynchronization takes place when the connection next becomes active or when the UOW is unshunted.

If recovery is outstanding, on completion of exchange lognames either resynchronization takes place or, in the case of a cold exchange, the PENDING condition is created.

SECURITYNAME (*data-area*)

Returns the security name of the remote system.

In a CICS system with security initialized (SEC=YES), and for an IPIC connection defined with LINKAUTH(SECUSER), the security name is used to establish the authority of the remote system.

The security name must be a valid RACF[®] user ID on this region. Access to protected resources on this region is based on the RACF user profile and its group membership. If the security name is not a valid RACF user ID when the IPCONN is installed, CICS uses the default user ID for the security name.

In a CICS system without security initialized (SEC=NO), or for an IPIC connection that is not defined with LINKAUTH(SECUSER), returns the value specified in the IP connection resource definition.

SENDCOUNT (*data-area*)

Returns a fullword binary value giving the number of SEND sessions defined for this IPIC connection.

SERVSTATUS (*cvda*)

Returns a CVDA value indicating whether data can be sent and received on the IPIC connection. CVDA values are as follows:

INSERVICE

Data can be sent and received.

OUTSERVICE

Data cannot be sent or received.

SSLTYPE (*cvda*)

Returns a CVDA value specifying the level of secure sockets support that is being used for this service. CVDA values are as follows:

NOSSL

The Secure Sockets Layer is not being used for this service.

SSL The Secure Sockets Layer without client authentication is being used for this service.

TCPIPSERVICE(*data-area*)

Returns the 8-character name of a PROTOCOL(IPIC) TCPIPSERVICE definition that defines the attributes of the inbound processing for this IPIC connection.

USERAUTH(*cvda*)

Returns a CVDA value that specifies the level of attach-time user security required for the connection. CVDA values are as follows:

LOCAL

CICS does not accept a user ID or password from clients. All requests will run under the link user ID.

IDENTIFY

Incoming attach requests must specify a user ID.

VERIFY

Incoming attach requests must specify a user ID and password.

DEFAULTUSER

CICS will not accept a user ID and password from the partner system. All requests run under the default user ID.

Conditions**END**

RESP2 values:

- 2 No more resource definitions of this type exist.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SYSIDERR

RESP2 values:

- 1 The IPIC connection cannot be found.

INQUIRE IPFACILITY

Retrieve information about an IP facility.

INQUIRE IPFACILITY

►►—INQUIRE IPFACILITY(*data-value*)—IPCONN(*data-area*)—IPFACILITYTYPE(*cvda*)————►►

Conditions:ILLOGIC, NOTAUTH, NOTFIND

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

This command is threadsafe

Description

The INQUIRE IPFACILITY command returns information about a particular IPFACILITY installed in your CICS system.

Options

IPFACILITY (*data-value*)

Specifies the 4-byte binary token identifying the IP facility to be queried. This should be one of the tokens returned in the IPFACILITIES list from an INQUIRE TASK command.

IPCONN (*data-area*)

Returns the 8-character value of the IPCONN with which this IP facility is associated.

IPFACILTYPE (*cvda*)

Returns a cvda value where:

PRINCIPAL

This is the task's principal facility.

ALTERNATE

This is the task's alternate facility.

Conditions

ILLOGIC

RESP2 values:

- 1 A START command has been issued when a browse of IPFACILITY resources is already in progress, or a NEXT or an END command has been issued when a browse of IPFACILITY resources is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFIND

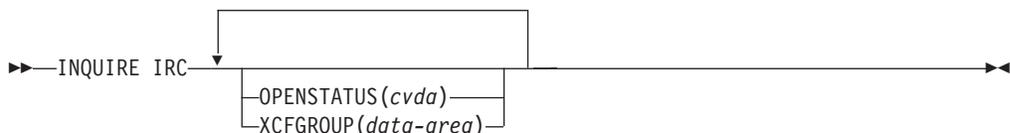
RESP2 values:

- 1 The named IPFACILITY cannot be found.

INQUIRE IRC

Show the IRC status.

INQUIRE IRC



Conditions: NOTAUTH

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The INQUIRE IRC command indicates whether interregion communication (IRC) is open, closed, or in a transitional state in your CICS system. IRC must be open for your CICS region to communicate with another CICS region using any of the multiregion operation (MRO) facilities (IRC, XM, or XCF).

Options

OPENSTATUS(*cvda*)

returns a CVDA value identifying the status of IRC in the system. CVDA values are:

CLOSED

IRC is closed for this system, or is not present in the system.

CLOSING

A SET IRC CLOSED request to quiesce MRO has been received; tasks that were already using an MRO link are being allowed to complete, but new tasks cannot use an MRO link.

IMMCLOSING

A SET IRC IMMCLOSE request to shut down MRO immediately has been received. Tasks that were using an MRO link are being terminated abnormally.

OPEN IRC is open for this system.

XCFGROUP(*data-area*)

returns the 8-character name of the cross-system coupling facility (XCF) group of which this region is a member.

If this region is not a member of an XCF group (because it has not signed on to IRC), XCFGROUP contains the XCF group the region would be in if XCF were opened.

For introductory information about XCF/MRO, see the *CICS Intercommunication Guide*.

Conditions

NOTAUTH

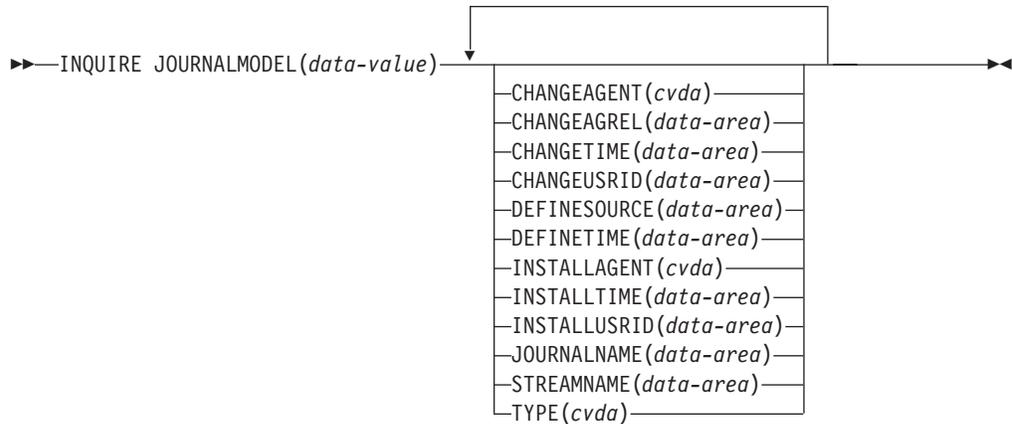
RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

INQUIRE JOURNALMODEL

Retrieve information about installed journal models, thus enabling you to obtain corresponding log stream names.

INQUIRE JOURNALMODEL



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

Description

The INQUIRE JOURNALMODEL command returns information about a particular installed journal model so that you can obtain corresponding log stream names.

Browsing

You can also browse through all of the journal model names on your system by using the browse options, START, NEXT, and END, on INQUIRE JOURNALMODEL commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

CICS returns journal models in alphanumeric sequence of the JOURNALNAMEs specified in the journal model, but with specific names being returned before the generic names. The following examples of journal names defined on journal models show the order in which the journal models are returned on a browse JOURNALMODEL operation:

```
DFHJ15
DFHJ25
DFHJ%0
DFH*
USERJNL1
USERJNL2
USERJNL*
```

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT(*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL(*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID(*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE(*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT** value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

INSTALLAGENT(*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

JOURNALMODEL (*data-value*)

Specifies the 1- to 8-character name of an installed journal model.

JOURNALNAME (*data-area*)

Returns the 1- to 8-character journal name, which can be a specific or a generic name. See the *CICS Resource Definition Guide* for further information about the JOURNALNAME operand.

STREAMNAME (*data-area*)

Returns the MVS log stream name (LSN) associated with the JOURNALMODEL entry.

The name can be a specific LSN or a template using a maximum of any three of the four symbols &USERID, &APPLID, &JNAME, and &SYSID.

The name, LSN or template, can be up to 26 characters in length. Names shorter than 26 characters are padded with trailing blanks (X'40').

TYPE (*cvda*)

Indicates the log stream type. The CVDA values are as follows:

DUMMY

Records are not written to any log stream.

MVS Records are written to an MVS log stream.

SMF Records are written to the MVS SMF log stream.

Conditions**END**

RESP2 values:

2 All authorized resources have been retrieved. All data areas specified on this command are left unchanged.

ILLOGIC

RESP2 values:

1 A START has been given when a browse is already in progress, or a NEXT or END has been given without a preceding START.

2 The browse token is not valid.

NOTAUTH

RESP2 values:

100 The user is not authorized for this command.

NOTFND

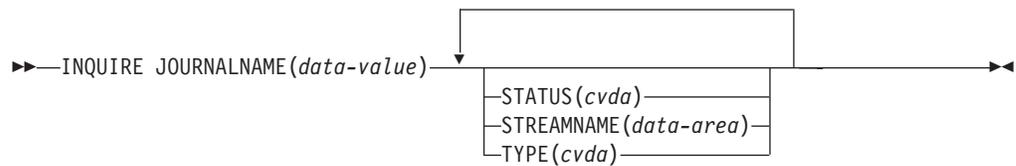
RESP2 values:

1 The specified journal model was not found.

INQUIRE JOURNALNAME

Retrieve information about the status of the system log and general logs.

INQUIRE JOURNALNAME



Conditions: END, ILLOGIC, JIDERR, NOTAUTH

Description

The INQUIRE JOURNALNAME command returns information about the journals (including the system log and general logs) on your system.

Browsing

You can also browse through all the journal entries in the journal names table on your system by using the browse options (START, NEXT, and END) on INQUIRE JOURNALNAME commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

Options

JOURNALNAME(*data-value*)

specifies a 1- to 8-character journal name.

To inquire on journals defined with a numeric identifier in the range 1–99, specify journal name DFHJ*nn*, where *nn* is the journal number.

To inquire on the system log, specify DFHLOG.

STATUS(*cvda*)

indicates the status of the journal. CVDA values are:

DISABLED

The journal has been disabled by a CEMT, or EXEC CICS, SET JOURNALNAME(...) command. It cannot be used until it is reenabled by the STATUS(ENABLED) or ACTION(RESET) options on a SET JOURNALNAME command.

ENABLED

The journal is installed and is available for use.

FAILED

The journal has experienced a log stream failure. It cannot be used until it is reenabled by the STATUS(ENABLED) or ACTION(RESET) options on a SET JOURNALNAME command, or until after the next CICS restart. The log stream should be deleted from the MVS system logger inventory before being used again.

STREAMNAME(*data-area*)

returns the MVS logger log stream name (LSN) associated with the journal name.

The name can be up to 26 characters in length. Names less than 26 character are padded with trailing blanks (X'40'). If the journal is defined by a journal model that specifies a type of DUMMY or SMF, CICS returns 26 blanks.

TYPE(*cvda*)

Indicates the type of log stream format. CVDA values are:

DUMMY

Records are not written to any log stream.

MVS Records are written to an MVS logger log stream.

SMF Records are written to the MVS SMF log stream.

Conditions**END**

RESP2 values:

2 All authorized resource definitions have been retrieved. All data areas specified on this command are left unchanged.

ILLOGIC

RESP2 values:

1 A START has been given when a browse is already in progress, or a NEXT, or an END, has been given without a preceding START.

JIDERR

RESP2 values:

1 The specified journal name was not found.

NOTAUTH

RESP2 values:

100 The user is not authorized for this command.

101 The user does not have the required access to the specified journal. (Not applicable to INQUIRE JOURNALNAME START, INQUIRE JOURNALNAME NEXT, or INQUIRE JOURNALNAME END commands.)

INQUIRE JOURNALNUM

This command is supported in releases of CICS earlier than CICS TS for OS/390 for retrieving information about the system log and user journals.

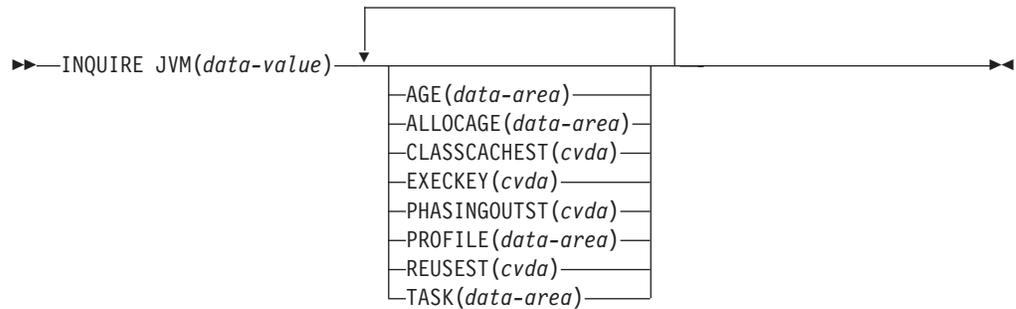
For this release, the command is replaced by the INQUIRE JOURNALNAME command. All the options on INQUIRE JOURNALNUM are obsolete, and the only run-time support provided by CICS for compatibility with earlier releases is to return the JIDERR exception condition. The translator translates the command, but issues a warning message.

The browse function is provided for compatibility with releases of CICS earlier than CICS TS for OS/390. A NORMAL condition is returned for the START browse and END browse operations. The END condition is returned for the NEXT browse operation.

INQUIRE JVM

Identify pooled JVMs in a CICS region and report their status.

INQUIRE JVM



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

Use the **INQUIRE JVM** command to inquire on the status of a particular pooled JVM by specifying its JVM number. You can also use the browse options to browse through all of the pooled JVMs in a CICS region to identify a JVM about which you want to inquire. You can use this command only for pooled JVMs. If you want to find out about a JVM server, use the **INQUIRE JVMSERVER** command.

Browsing

You can browse through all of the pooled JVMs in a CICS region by using the browse options (START, NEXT, and END) on the **INQUIRE JVM** command. The browse starts at the first JVM in the JVM pool, and moves through them in order of their JVM numbers. The browse guarantees that data for each JVM that exists before the first **INQUIRE JVM NEXT** command, and still exists after the last **INQUIRE JVM NEXT** command, is returned on exactly one **INQUIRE JVM NEXT** call. JVMs that are started while the browse is in progress can be found by the browse. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

Options

JVM(data-value)

Specifies or returns the JVM number, a fullword binary value that uniquely identifies the JVM. This information is returned by an **INQUIRE JVM** browse, and you can then specify the JVM number for the JVM about which you want to inquire. The JVM number used in CICS is the same number that is used to identify the JVM in the z/OS UNIX environment, where it is known as the process ID (PID) for the JVM.

AGE(data-area)

Returns a fullword binary value giving the number of seconds since the JVM was initialized.

ALLOCAGE (*data-area*)

Returns a fullword binary value giving the number of seconds for which the JVM has been allocated to its task or zero if the JVM is not currently allocated to a task.

CLASSCACHST (*cvda*)

Returns a CVDA value indicating whether the JVM uses the shared class cache. This option has the following CVDA values:

CLASSCACHE

The JVM profile for this JVM specified the use of the shared class cache.

NOCLASSCACHE

The JVM profile for this JVM did not specify the use of the shared class cache.

EXECKEY (*cvda*)

Returns a CVDA value indicating the EXECKEY of the JVM. This option has the following CVDA values:

CICSEXECKEY

The JVM runs in CICS key.

USEREXECKEY

The JVM runs in user key.

PHASINGOUTST (*cvda*)

Returns a CVDA value indicating whether the JVM is being phased out because the JVM pool or class cache has been terminated. This option has the following CVDA values:

PHASEOUT

The JVM is being phased out.

NOPHASEOUT

The JVM is not being phased out. It is available for allocation, or will be available for allocation when the current allocation is ended.

PROFILE (*data-area*)

Returns the 8-character name of the profile used to initialize this JVM.

REUSEST (*cvda*)

Returns a CVDA value indicating whether this JVM can be reused. This option has the following CVDA values:

REUSE

The JVM profile for this JVM specified that it is continuous.

NOREUSE

The JVM profile for this JVM specified that it is single-use.

TASK (*data-area*)

Returns a 4-byte packed decimal value giving the task to which the JVM is allocated, or zero if the JVM is not currently allocated to a task.

Conditions**END**

RESP2 values:

2 There are no more JVMs to browse.

ILLOGIC

RESP2 values:

- 1 A **START** command has been given when a browse of JVMs is already in progress. Alternatively, a **NEXT** or an **END** command has been given without a preceding **START** command.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

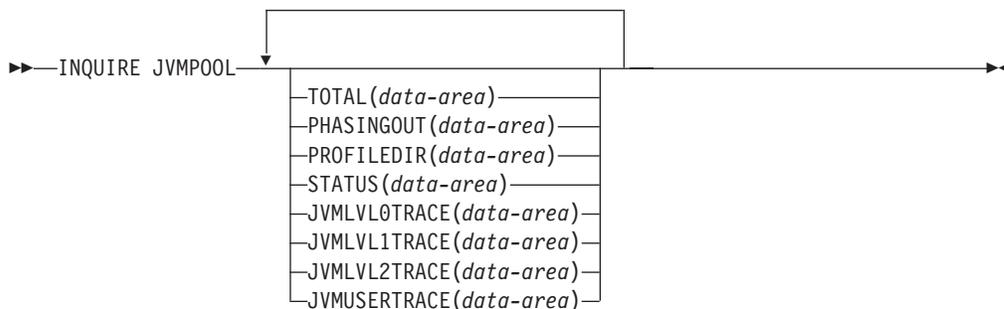
RESP2 values:

- 4 The named JVM was not found.

INQUIRE JVMPOOL

Retrieve information about the pool of JVMs in the CICS address space.

INQUIRE JVMPOOL



Conditions: NOTAUTH

This command is threadsafe.

For more information about the use of CVDAAs, see CICS-value data areas (CVDAAs) in CICS System Programming Reference.

Description

The **INQUIRE JVMPOOL** command returns information about the pool of JVMs in the CICS region. Only one pool of JVMs exists in a CICS region and therefore no name or identifier is required on this command. This command does not return information about JVM servers. If you want to find out about a JVM server, use the **INQUIRE JVMSERVER** command.

CICS determines the information you request from the pooled JVMs that are started and pre-initialized in the CICS region.

Options

JVMLVL0TRACE (*data-area*)

JVMLVL1TRACE (*data-area*)

JVMLVL2TRACE (*data-area*)

JVMUSERTRACE (*data-area*)

Each of these options return a 240-character data value giving the current options for JVM tracing. These options are used when the associated level of

tracing has been activated using the CICS-supplied transaction CETR, the **SET TRACETYPE** command, or the **SPCTRSJ** system initialization parameter. Trace levels 29-32 for the SJ component correspond to JVMLVL0TRACE, JVMLVL1TRACE, JVMLVL2TRACE, and JVMUSERTRACE.

PHASINGOUT(*data-area*)

Returns a fullword binary field giving the number of JVMs that are marked for removal from the JVM pool. These JVMs are still allocated to a task that is currently running, or has run, a Java program in the JVM.

PROFILEDIR(*data-area*)

Returns a 240-character data value of the directory on z/OS UNIX that contains the JVM profiles for CICS. This value is taken from the **JVMPROFILEDIR** system initialization parameter.

STATUS(*cvda*)

Returns a CVDA value indicating the overall status of the JVM pool. This option has the following CVDA values:

ENABLED

The pool is enabled for use and Java programs can run using JVMs from the pool. This option is the normal status.

DISABLED

The pool is disabled and new requests cannot be serviced from the pool. Programs can still be running if they were started before the JVM pool was disabled.

TOTAL(*data-area*)

Returns a fullword binary field giving the number of JVMs that have been initialized and are available for use or allocated to tasks. This total includes JVMs that are in the process of being terminated and removed from the region and included on the PHASINGOUT count.

Conditions

NOTAUTH

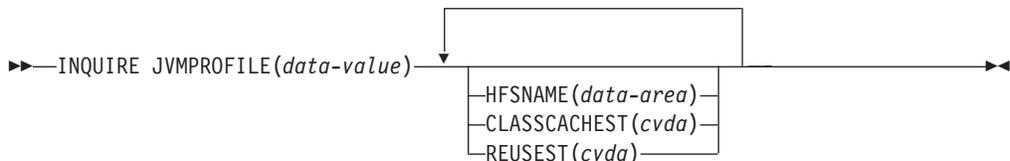
RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

INQUIRE JVMPROFILE

Identify JVM profiles that have been used in a CICS region for pooled JVMs, and report their full path name and association with the shared class cache.

INQUIRE JVMPROFILE



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The **INQUIRE JVMPROFILE** command retrieves the full path name of the z/OS UNIX file for a JVM profile, and tells you whether the pooled JVM associated with this profile uses the shared class cache.

INQUIRE JVMPROFILE can find only JVM profiles that have been used during the lifetime of the CICS region, for pooled JVMs that can run applications. It does not find JVM profiles that have not been used or those profiles that are used for a JVM server. To find out what JVM profile is used for a JVM server, use the **INQUIRE JVMSERVER** command.

Browsing

You can browse through all the JVM profiles that have been used in a CICS region by using the browse options (**START**, **NEXT**, and **END**) on **INQUIRE JVMPROFILE** commands. **INQUIRE JVMPROFILE** commands do not support the **AT** option for browsing. In browse mode, the JVM profiles are returned in alphabetic order. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

Options

JVMPROFILE (*data-value*)

Specifies the 8-character name of the JVM profile, as used in a **PROGRAM** resource.

You must enter the profile name using the same combination of upper and lower case characters that are present in the z/OS UNIX file name.

HFSNAME (*data-area*)

Returns a 255-character area containing the full path name of the z/OS UNIX file for the JVM profile.

CLASSCACHEST (*cvda*)

Returns a CVDA value indicating whether pooled JVMs that use this JVM profile use the shared class cache. This option has the following CVDA values:

CLASSCACHE

The JVM profile specifies the use of the shared class cache.

NOCLASSCACHE

The JVM profile does not specify the use of the shared class cache.

REUSEST (*cvda*)

Returns a CVDA value indicating whether pooled JVMs that are created using this JVM profile can be reused. This option has the following CVDA values:

REUSE

Pooled JVMs with this JVM profile are continuous.

NOREUSE

Pooled JVMs with this JVM profile are single-use.

Conditions

END

RESP2 values:

2 There are no more JVM profiles to browse.

ILLOGIC

RESP2 values:

1 A START has been given when a browse of JVM profiles is already in progress, or a NEXT or an END has been given without a preceding START.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

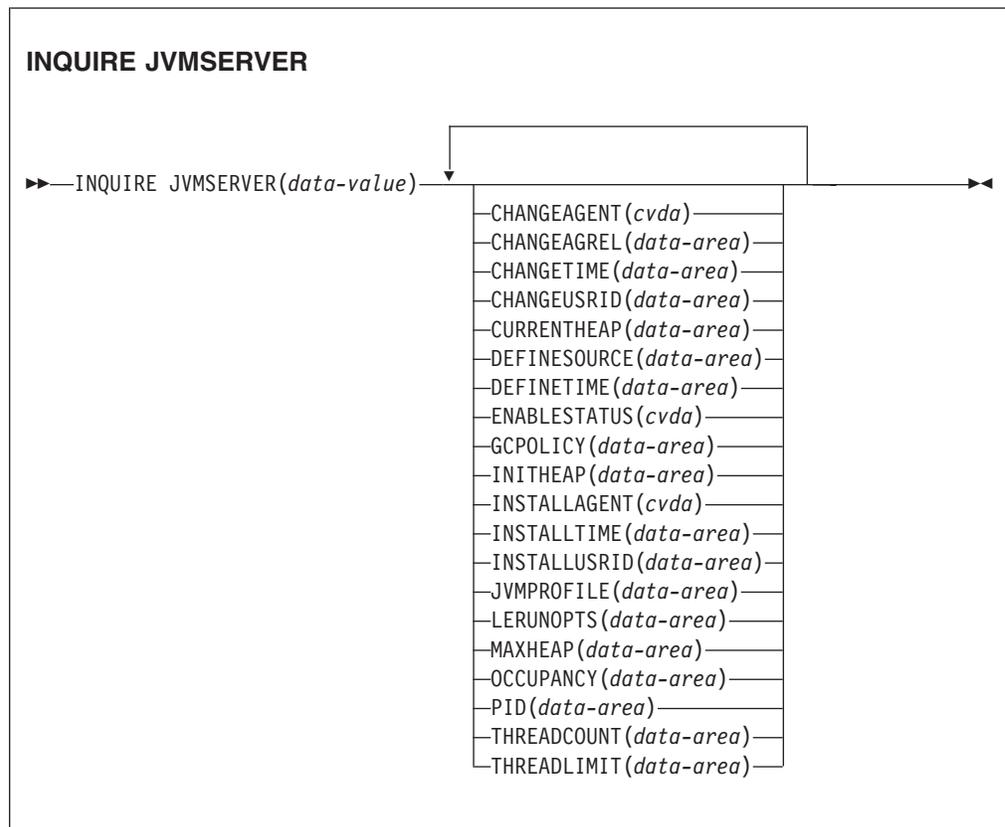
NOTFND

RESP2 values:

3 The named JVM profile was not found.

INQUIRE JVMSERVER

Retrieve information about the JVM server runtime environment in the CICS region.



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.

Description

The **INQUIRE JVMSERVER** command retrieves information about one or more JVM servers that are running in the CICS region.

Browsing

You can browse through all the JVMSERVER definitions installed in the region, using the browse options (START, NEXT, and END) on INQUIRE JVMSERVER commands.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT(*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL(*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see FORMATTIME in CICS Application Programming.

CHANGEUSRID(*data-area*)

Returns the 8-character user ID that ran the change agent.

CURRENTHEAP(*data-area*)

Returns a doubleword binary value indicating the current size of the heap in bytes that is allocated to the JVM server.

| **DEFINESOURCE**(*data-area*)

| Returns the 8-character source of the resource definition. The **DEFINESOURCE**
| value depends on the **CHANGEAGENT** value. For more information, see
| Summary of the resource signature field values in the Resource Definition
| Guide.

| **DEFINETIME**(*data-area*)

| Returns an **ABSTIME** value that represents the time stamp when the resource
| definition was created.

| **ENABLESTATUS**(*cvda*)

| Returns a **CVDA** value indicating the overall status of the JVM server. The
| **CVDA** values are as follows:

| **ENABLED**

| The JVM server has started and is enabled for use.

| **ENABLING**

| The JVM server is starting.

| **DISABLED**

| The JVM server is stopped and any new requests cannot be processed.

| **DISABLING**

| The JVM server is stopping. Threads can still be running if they were
| started before the JVM server was stopped.

| **DISCARDING**

| The **JVMSERVER** resource is being discarded.

| **GCPOLICY**(*data-area*)

| Returns a 32-character value indicating the garbage collection policy that is
| being used by the JVM server.

| **INITHEAP**(*data-area*)

| Returns a doubleword binary value that indicates the initial size of the heap in
| bytes that is allocated to the JVM server. This value is set by the **-Xms** option in
| the JVM profile.

| **INSTALLAGENT**(*cvda*)

| Returns a **CVDA** value that identifies the agent that installed the resource. The
| possible values are as follows:

| **CREATESPI**

| The resource was installed by an **EXEC CICS CREATE** command.

| **CSDAPI**

| The resource was installed by a **CEDA** transaction or the
| programmable interface to **DFHEDAP**.

| **GRPLIST**

| The resource was installed by **GRPLIST INSTALL**.

| **INSTALLTIME**(*data-area*)

| Returns an **ABSTIME** value that represents the time stamp when the resource
| was installed.

| **INSTALLUSRID**(*data-area*)

| Returns the 8-character user ID that installed the resource.

| **JVMPROFILE**(*data-area*)

| Returns the 1-8 character profile name of the JVM server.

JVMSEVER(*data-value*)

Specifies the name of the JVMSEVER resource about which you are inquiring. The name can be up to 8 characters in length.

LERUNOPTS(*data-area*)

Returns the 1-8 character name of the program that defines the runtime options for the Language Environment enclave.

MAXHEAP(*data-area*)

Returns a doubleword binary value that indicates the maximum size of the heap in bytes that is allocated to the JVM server. This value is set by the **-Xmx** option in the JVM profile.

OCCUPANCY(*data-area*)

Returns a doubleword binary value that indicates the size of the heap in bytes after the last garbage collection ran in the JVM server.

PID(*data-area*)

Returns a fullword value that indicates the process ID (PID) of the JVM.

THREADCOUNT(*data-area*)

Returns a fullword binary value giving the number of threads that are currently running inside the JVM server.

THREADLIMIT(*data-area*)

Returns a fullword binary value giving the number of threads that are allowed in the Language Environment enclave for the JVM server. Each thread runs under a T8 TCB.

Conditions**END**

RESP2 values:

2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

NOTFND

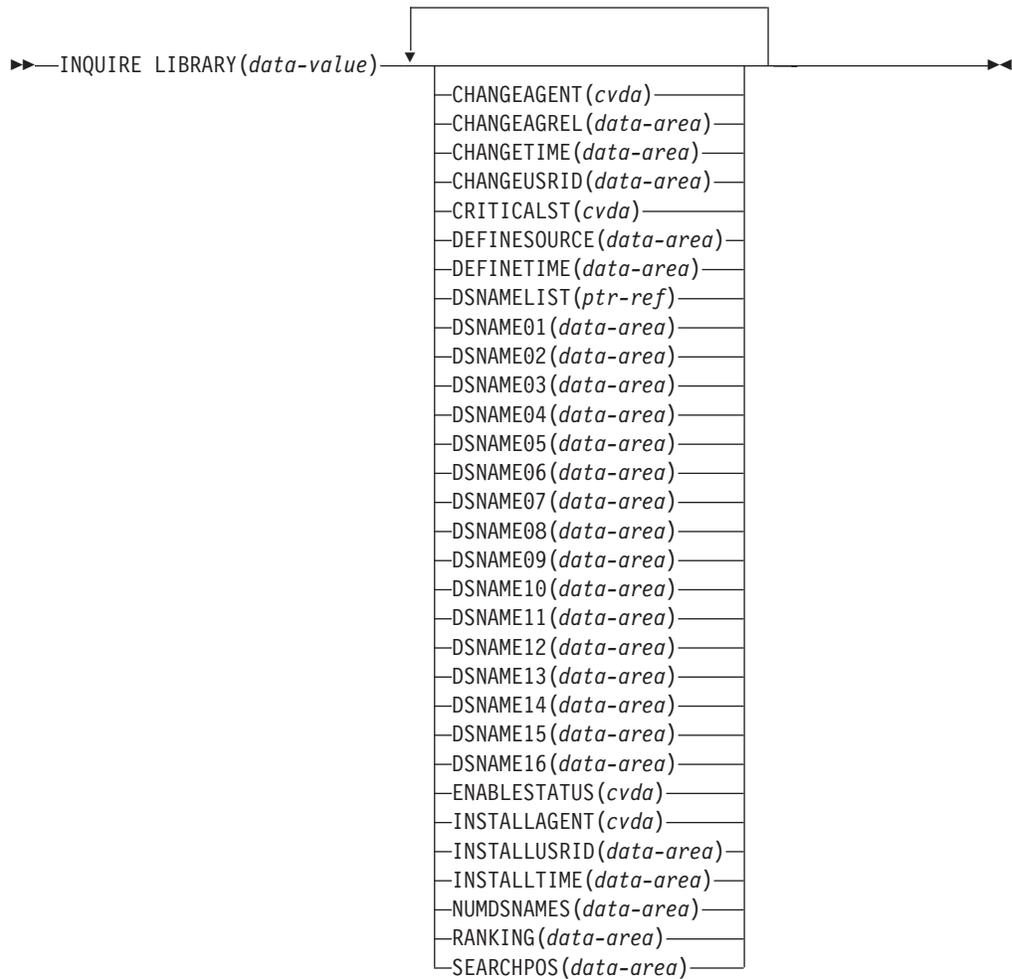
RESP2 values:

3 The JVMSEVER resource cannot be found.

INQUIRE LIBRARY

Retrieve information about a LIBRARY.

INQUIRE LIBRARY



Conditions: END, ILLOGIC, NOTAUTH, NOTFIND

For more information about the use of CVDAs, see “CICS-value data areas (CVDA)” on page 8.

This command is threadsafe

Description

The `INQUIRE LIBRARY` command returns information about a particular library installed in your CICS system.

Browsing

You can also browse through the library resources in your system by using the browse options, `START`, `NEXT`, and `END`, on `INQUIRE LIBRARY` commands. In browse mode, the library resources are returned in search order, starting with the first library in the search order. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT(*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

SYSTEM

The resource definition was last changed by the CICS or CICSplex system.

CHANGEAGREL(*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see FORMATTIME in CICS Application Programming.

CHANGEUSRID(*data-area*)

Returns the 8-character user ID that ran the change agent.

CRITICALST(*cvda*)

Returns a CVDA value identifying whether the library is critical to the start up of CICS. CVDA values are as follows:

CRITICAL

The LIBRARY is critical to CICS startup. If the library cannot be successfully installed during CICS startup for any reason, then a G0 or CANCEL message is issued. The operator can decide whether to override the critical status and allow CICS to start or not. If CICS is allowed to continue, the library is installed in a DISABLED status, unless install was not possible at all; for example, because of a short-on-storage condition.

If the reply is to continue with the startup, the library is not recatalogued as NONCRITICAL, so the critical status must be explicitly set to NONCRITICAL you decide that the library is not to be regarded as CRITICAL in future.

NONCRITICAL

The library is not critical to CICS startup. If the library cannot be successfully installed during CICS startup, the library is left in an installed but disabled state. If installation is not possible for the LIBRARY, a warning message is issued and CICS startup continues.

LIBRARY (*data-value*)

Specifies the 8-character name of the LIBRARY about which you are inquiring.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

DSNAMELIST (*ptr-ref*)

Returns the address of a buffer containing all the data sets in the LIBRARY concatenation. The buffer contains an array of 44-character data set names. This buffer is intended for use when the DFHRPL concatenation contains more than 16 data sets. The number of data sets is indicated by the NUMDSNAMES parameter, but, if the library is not DFHRPL, some of the slots in the list will be empty if the data set names at those positions are not specified on the dynamic LIBRARY definition.

DSNAME01-16 (*data-area*)

Returns the 44-character names of data sets in the library concatenation. If this library is a dynamically defined, these are the data sets specified on the LIBRARY definition, all but one of which can be blank. If this DFHRPL is the statically defined, then these are the first 16 data sets in the DFHRPL concatenation, or as many data sets as are specified up to 16, with the remaining DSNAME_{xx} fields being blank. The DFHRPL concatenation contains more than 16 data sets, you can use the DSNAMELIST option to obtain all of the data sets.

ENABLESTATUS (*cvda*)

Returns a CVDA value identifying whether the library is currently included in the overall library search order. CVDA values are as follows:

DISABLED

The LIBRARY is disabled, and is not currently included in the library search order. The data sets in this LIBRARY concatenation are not searched for program artifacts to load.

DISABLING

A request to disable the library has been received, but is still being processed.

ENABLED

The library is enabled, and is currently included in the library search order. The data sets in this library concatenation will be searched for program artifacts to load.

ENABLING

A request to enable the library has been received, but is still being processed.

DISCARDING

A request to discard the LIBRARY from the CICS system has been received, but is still being processed.

INSTALLAGENT(*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

SYSTEM

The resource was installed by the CICS or CICSplex SM system.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

NUMDSNAMES(*data-area*)

Returns a fullword binary value containing the number of data sets in the library concatenation. For a dynamically defined library, this value is the number of non blank DSNAMExx values, and cannot be a value larger than 16. For the statically defined DFHRPL, this value is the number of data sets in the concatenation, and can be a value larger than 16.

RANKING(*data-area*)

Returns a fullword binary value that indicates where this library appears in the overall LIBRARY search order relative to other LIBRARY concatenations. A lower number indicates that this LIBRARY is searched for programs to load before other LIBRARY resources with higher ranking numbers. Libraries appear in the search order, in order of ranking. However libraries of equal RANKING appear in the search order in the order in which they were installed or created in the local CICS system, with a library that was installed earlier appearing before one that was installed later.

SEARCHPOS(*data-area*)

Returns a fullword binary value containing the current absolute position of this library in the overall LIBRARY search order. The first enabled library in the search order has a SEARCHPOS of 1, the next enabled library will have a SEARCHPOS of 2, and so on. The SEARCHPOS is not the same as the RANKING, although its value is determined by the relative ranking values of the various library resources in the system.

The SEARCHPOS values, relative to other library resources with the same RANKING value, are related to installation or create time, but their SEARCHPOS values relative to each other are retained across a warm or

emergency restart. There is no guarantee that the relative SEARCHPOS values of library resources with the same RANKING will be the same after a cold or initial start.

If the library is disabled, the SEARCHPOS is 0, indicating that the library does not participate in the overall search.

Conditions

END

RESP2 values:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 A START command has been issued when a browse of library resources is already in progress, or a NEXT or an END command has been issued when a browse of library resources is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFIND

RESP2 values:

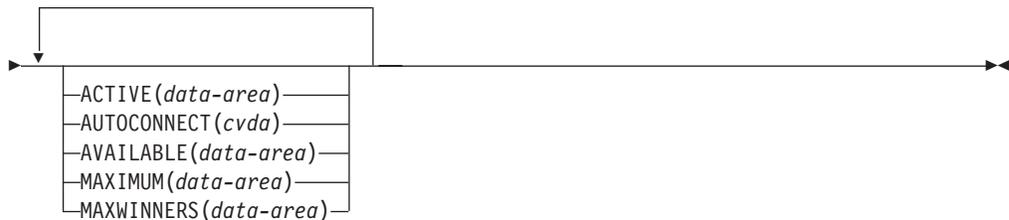
- 1 The named library cannot be found.

INQUIRE MODENAME

Retrieve information about a session group within a connection.

INQUIRE MODENAME

►►—INQUIRE MODENAME(*data-value*)—CONNECTION(*data-value*)—►►



Conditions: END, ILLOGIC, NOTAUTH, SYSIDERR

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The **INQUIRE MODENAME** command returns information about a group of sessions (sometimes called a “mode”) that has been defined within a connection to a remote system. (The MODENAME for the group is the name assigned to the SESSIONS resource definition that creates it.)

MODENAMES are unique within a given connection, but not across connections. Therefore, to look at a particular session group, you must specify data values for both the MODENAME and CONNECTION options.

Browsing

You can also browse through all of the session groups for a particular connection, or all groups for all connections, by using the browse options (START, NEXT, and END) on **INQUIRE MODENAME** commands.

As in a single **INQUIRE MODENAME** command, you must include both the MODENAME and CONNECTION options on an **INQUIRE MODENAME NEXT** command. The data-value for MODENAME is optional; if you provide it, CICS uses it to return the name of the session group. If you want to limit your browse to a single connection specify the data-value for the CONNECTION.

To see all groups, initialize the CONNECTION data-value before running the **INQUIRE MODENAME NEXT** command. Use the output from the CONNECTION data-value on each **INQUIRE MODENAME NEXT** command as the input to the next **INQUIRE MODENAME NEXT** command. You can then browse through all the modenames of all connections.

See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

Options

ACTIVE(*data-area*)

Returns a halfword binary field giving the number of sessions within the group that are currently in use.

AUTOCONNECT(*cvda*)

Returns a CVDA value indicating whether the sessions within this group are to be bound automatically whenever CICS starts communication with z/OS Communications Server. CVDA values are:

ALLCONN

CICS tries to bind both contention-winner and contention-loser sessions.

AUTOCONN

CICS tries to bind only sessions for which it is contention winner.

NONAUTOCONN

CICS does not try to bind any sessions.

AVAILABLE(*data-area*)

Returns a halfword binary field giving the current number of sessions in the group (the number “bound”).

CONNECTION (*data-value*)

Specifies the 4-character identifier of the remote system with which this group of sessions is associated (the name of the CONNECTION resource definition for that system).

MAXIMUM (*data-area*)

Returns a halfword binary field giving the maximum number of sessions that the definition of the session group permits.

MAXWINNERS (*data-area*)

Returns a halfword binary field giving the maximum number of sessions that the definition of the session group permits to be contention winners. A single-session APPC definition installed by RDO or autoinstall always shows 0 for this field.

MODENAME (*data-value*)

Specifies the 8-character identifier of the group of sessions about which you are inquiring. Modename is the name of the SESSIONS resource definition for the group.

Conditions**END**

RESP2 values:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SYSIDERR

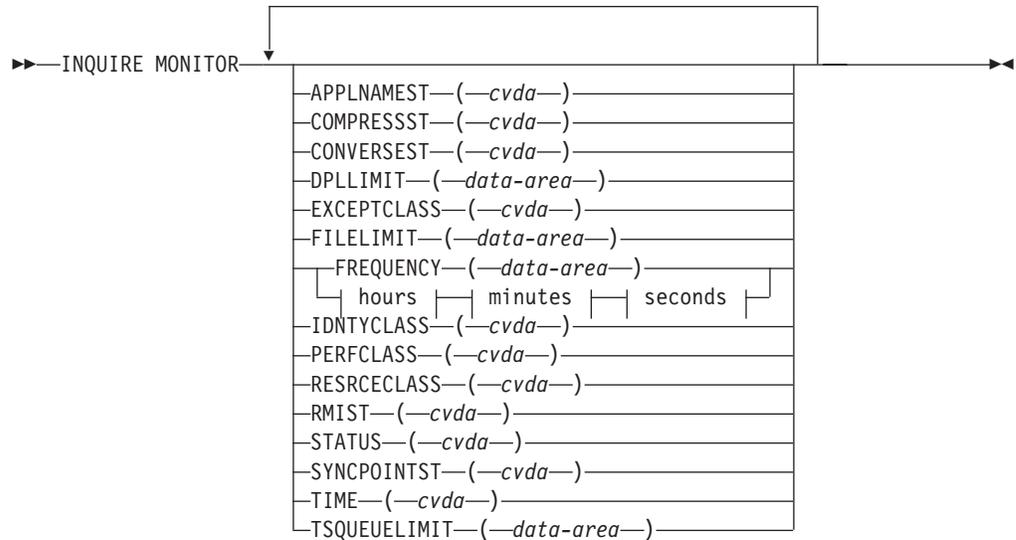
RESP2 values:

- 1 The connection cannot be found.
- 2 The modename within the connection cannot be found.
- 3 The connection specified on an **INQUIRE MODENAME NEXT** cannot be found.

INQUIRE MONITOR

Retrieve the status of CICS monitoring.

INQUIRE MONITOR



hours:

|—FREQUENCYHRS—(—data-area—)|

minutes:

|—FREQUENCYMINS—(—data-area—)|

seconds:

|—FREQUENCYSECS—(—data-area—)|

Conditions: NOTAUTH

For more information about the use of CVDAs, see CICS-value data areas (CVDAs) in CICS System Programming Reference.

Description

Use the INQUIRE MONITOR command to find out whether CICS monitoring is active, which types of data are being recorded, and other recording options.

CICS monitoring is controlled by a master switch (the STATUS option) and four switches that govern which types of data are recorded (the EXCEPTCLASS, PERFCLASS, RESRCECLASS, and IDNTYCLASS options). See the “SET MONITOR” on page 693 command for a description of monitor data classes and details about how the switches interact.

Options

APPLNAMEST (*cvda*)

Returns a CVDA value indicating whether CICS application naming support is enabled. CVDA values are as follows:

APPLNAME

CICS application naming support is enabled.

NOAPPLNAME

CICS application naming support is not enabled.

COMPRESSST (*cvda*)

Returns a CVDA value indicating whether data compression is active for the CICS SMF 110 monitoring records produced by the CICS monitoring facility. CVDA values are as follows:

COMPRESS

Data compression is being performed for the monitoring records. Data compression is the default.

NOCOMPRESS

Data compression is not being performed for the monitoring records.

CONVERSEST (*cvda*)

Returns a CVDA value indicating how CICS is to record performance data for conversational tasks (tasks that wait for terminal or session input). CVDA values are as follows:

CONVERSE

CICS produces a performance class record for a conversational task each time it waits for terminal input as well as at task end, representing the part of the task since the previous terminal wait or task start. These waits occur during execution of a CONVERSE command or a RECEIVE command that follows a SEND.

NOCONVERSE

CICS accumulates performance data across terminal waits and produces a single performance class record for a conversational task.

DPLLIMIT (*data-area*)

Returns the maximum number of distributed program link requests for which CICS is to perform transaction resource monitoring.

EXCEPTCLASS (*cvda*)

Returns a CVDA value indicating whether the exception class of monitoring data is recorded when monitoring is active. CVDA values are as follows:

EXCEPT

Exception data is recorded.

NOEXCEPT

Exception data is not recorded.

FILELIMIT (*data-area*)

Returns the maximum number of files for which CICS is to perform transaction resource monitoring.

FREQUENCY (*data-area*)

Returns the interval at which CICS produces performance class records for long-running tasks. If a task runs longer than the FREQUENCY interval, CICS records its performance data separately for each interval or fraction.

The frequency interval has two formats:

- A composite (packed decimal format 0hhmmss+, 4 bytes long), which you obtain by using the FREQUENCY option.
- Separate hours, minutes, and seconds, which you obtain by specifying the FREQUENCYHRS, FREQUENCYMIN, and FREQUENCYSEC options.

A value of zero indicates that frequency reporting is inactive; that is, recording of performance data is not affected by the duration of the task.

FREQUENCYHRS(*data-area*)

Returns the hours component of the frequency interval, in fullword binary form. See the FREQUENCY option.

FREQUENCYMIN(*data-area*)

Returns the minutes component of the frequency interval, in fullword binary form. See the FREQUENCY option.

FREQUENCYSEC(*data-area*)

Returns the seconds component of the frequency interval, in fullword binary form. See the FREQUENCY option.

IDNTYCLASS(*cvda*)

Returns a CVDA value indicating whether the identity class of monitoring data is recorded when monitoring is active. CVDA values are as follows:

IDNTY

Identity data is recorded.

NOIDNTY

Identity data is not recorded.

PERFCLASS(*cvda*)

Returns a CVDA value indicating whether the performance class of monitoring data is recorded when monitoring is active. CVDA values are as follows:

NOPERF

Performance data is not recorded.

PERF Performance data is recorded.

RESRCECLASS(*cvda*)

Returns a CVDA value indicating whether transaction resource monitoring is active in the CICS region. CVDA values are as follows:

NORESRCE

Transaction resource monitoring is not active.

RESRCE

Transaction resource monitoring is active.

RMIST(*cvda*)

Returns a CVDA value indicating whether additional performance monitoring is active for the resource managers used by your transactions. CVDA values are as follows:

RMI Performance monitoring is active for the resource managers used by your transactions.

NORMI

Performance monitoring is not active for the resource managers used by your transactions.

STATUS(*cvda*)

Returns a CVDA value identifying whether CICS monitoring is active in the region. CVDA values are as follows:

OFF CICS monitoring is not active in the region. No monitoring data is accumulated or written out, regardless of the settings of the monitoring data classes.

ON CICS monitoring is active. Data is accumulated for all classes of monitor data and is written out for those classes that are active.

SYNCPOINTST(*cvda*)

Returns a CVDA value indicating whether CICS records performance class data separately for each unit of work (UOW) in tasks that contain multiple UOWs. A UOW in a task ends when a sync point occurs, either explicitly (a SYNCPOINT command) or implicitly (a DL/I TERM call, for example, or task end). A new UOW begins immediately after, except at end of task. When rollback occurs on a sync point, the UOW does not end. CVDA values are as follows:

NOSYNCPOINT

Performance data is combined over all UOWs in a task for recording.

SYNCPOINT

Performance data is recorded separately for each UOW.

TIME(*cvda*)

Returns a CVDA value identifying whether the performance class time-stamp fields returned to an application using the COLLECT STATISTICS MONITOR command are expressed in local or Greenwich mean time. The value of this option has no effect on the other classes of monitoring data. See the *CICS Customization Guide* for information on the SMF header. CVDA values are as follows:

GMT Time stamps are Greenwich mean time.

LOCAL

Time stamps are local time.

TSQUEUELIMIT(*data-area*)

Returns the maximum number of temporary storage queues for which CICS is to perform transaction resource monitoring.

Conditions

NOTAUTH

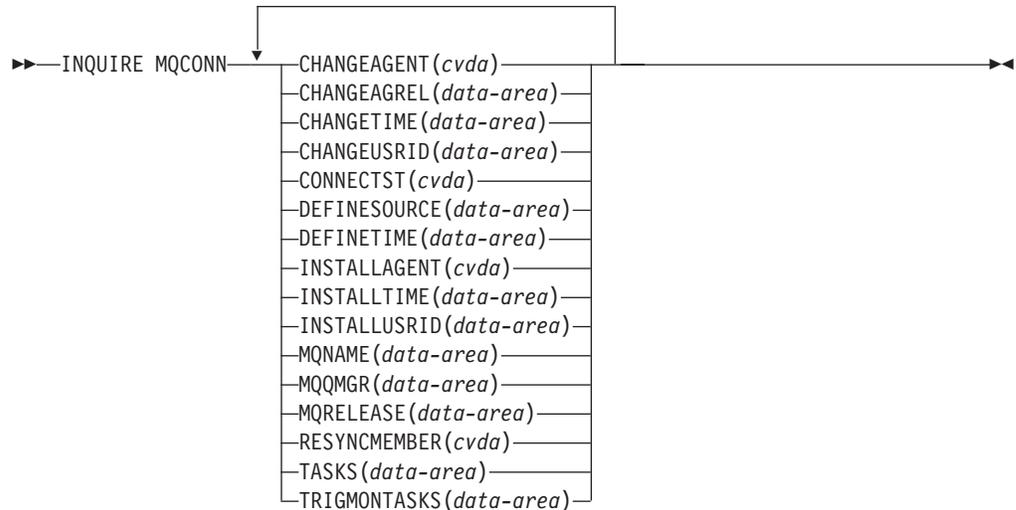
RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

INQUIRE MQCONN

Inquire on the attributes and status of the connection between CICS and WebSphere MQ.

INQUIRE MQCONN



Conditions: NOTAUTH, NOTFND

This command is threadsafe.

Description

The **INQUIRE MQCONN** command returns information about attributes of the currently installed MQCONN resource definition, which defines the connection to WebSphere MQ, and about the status of the connection.

Because only one MQCONN resource definition can be installed at a time, the name of the MQCONN resource definition is not required on input.

This command does not inquire on the INITQNAME attribute of the MQCONN resource definition, which specifies the name of the initiation queue. Use the **INQUIRE MQINI** command to inquire on the initiation queue name.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see FORMATTIME in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

CONNECTST (*cvda*)

Returns the status of the CICS-MQ connection. CVDA values are as follows:

CONNECTED

CICS is connected to WebSphere MQ.

NOTCONNECTED

CICS is not connected to WebSphere MQ.

CONNECTING

CICS is currently attempting to connect to WebSphere MQ.

DISCONNING

CICS is currently disconnecting from WebSphere MQ.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

MQNAME(*data-area*)

Returns the 1 - 4 character name of the WebSphere MQ queue manager or queue-sharing group that you specified in the MQCONN resource definition (or using a SET MQCONN command) for the CICS region.

MQQMGR(*data-area*)

Returns the 1 - 4 character name of the WebSphere MQ queue manager to which CICS is connected, or to which CICS is waiting to connect.

- If CICS is connected to WebSphere MQ, this field shows the name of the queue manager to which CICS is connected. If you specified a queue-sharing group in the MQCONN resource definition for the CICS region, the queue manager shown here is the one that was chosen from the group.
- If CICS is not connected to WebSphere MQ, this field usually contains blanks. However, if you specified a queue-sharing group in the MQCONN resource definition for the CICS region, and CICS is waiting to reconnect to a specific queue manager in the queue-sharing group because it is holding outstanding units of work for that queue manager, the name of the specific queue manager is shown, and the status of the connection is shown as CONNECTING. For this situation to arise, the RESYNCMEMBER attribute in the MQCONN resource definition must specify resynchronization.

MQRELEASE(*data-area*)

If CICS is connected to WebSphere MQ, this option returns the 4-digit release number of WebSphere MQ; for example, 0600. When CICS is not connected to WebSphere MQ, MQRELEASE returns blanks.

RESYNCMEMBER(*cvda*)

This option applies only if you have specified a queue-sharing group for the CICS-MQ connection. It shows the strategy that CICS adopts if outstanding units of work are being held for the last queue manager to which CICS was connected from the queue-sharing group. Units of work that are shunted indoubt are not included in this process, because CICS itself cannot resolve those units of work at this time. Resynchronization for those UOWs occurs when CICS has resynchronized with its remote coordinator. CVDA values are as follows:

RESYNC

CICS connects to the same queue manager.

NORESYNC

CICS makes one attempt to connect to the same queue manager. If that attempt fails, CICS connects to any member of the queue-sharing group and issues a warning message about the outstanding units of work.

GROUPRESYNC

CICS connects to any member of the queue-sharing group. The queue manager is chosen by WebSphere MQ and it asks CICS to resolve indoubt units of work on behalf of all eligible queue managers in the queue-sharing group. This function is called *group unit of recovery*.

NOTAPPLIC

A queue-sharing group is not specified for the CICS-MQ connection.

TASKS (*data-area*)

Returns the current number of tasks that are using the CICS-MQ connection, including trigger monitor tasks, as a fullword binary value.

TRIGMONTASKS (*data-area*)

Returns the current number of trigger monitor tasks that are using the CICS-MQ connection, as a fullword binary value.

Conditions

NOTFND

RESP2 values:

1 The MQCONN resource definition cannot be found.

NOTAUTH

RESP2 values:

100 Command authorization failure.

INQUIRE MQINI

Inquire on the name of the default initiation queue to be used for the connection between CICS and WebSphere MQ.

INQUIRE MQINI

```
▶▶—INQUIRE MQINI—(—data-value—)—CHANGEAGENT(cvda)—CHANGEAGREL(data-area)—▶▶
▶—CHANGETIME(data-area)—CHANGEUSRID(data-area)—DEFINESOURCE(data-area)—▶▶
▶—DEFINETIME(data-area)—INITQNAME(data-area)—INSTALLAGENT(cvda)—▶▶
▶—INSTALLTIME(data-area)—INSTALLUSRID(data-area)—▶▶▶▶
```

Conditions: NOTAUTH, NOTFND

This command is threadsafe.

Description

The INQUIRE MQINI command inquires on the name of the default initiation queue used for the connection between CICS and WebSphere MQ.

The MQINI resource represents the default initiation queue. MQINI is an implicit resource that exists when you install an MQCONN resource definition in the CICS region with the INITQNAME attribute specified. (Only one MQCONN resource definition can be installed at a time.) The name of the MQINI resource is DFHMQINI. You must specify this resource name on the command.

If you want to change the MQINI resource definition, you must reinstall the MQCONN resource definition with an appropriate MQINI attribute.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are

CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

AUTOINSTALL

The resource was autoinstalled as a result of specifying an initiation queue name on a CKQC START command, and the previously installed MQCONN definition did not specify a value for INITQNAME.

DYNAMIC

The resource was defined as a result of an MQCONN resource definition with INITQNAME specified.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see FORMATTIME in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

INITQNAME (*data-area*)

Returns the 1- to 48-character name of the default initiation queue that is represented by the MQINI resource.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

AUTOINSTALL

The resource was autoinstalled as a result of specifying an initiation queue name on a CKQC START command, and the previously installed MQCONN definition did not specify a value for INITQNAME.

DYNAMIC

The resource was installed as a result of the installation of an MQCONN with INITQNAME specified.

INSTALLTIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID (*data-area*)

Returns the 8-character user ID that installed the resource.

Conditions**NOTFND**

RESP2 values:

- 1 The MQCONN resource definition that implies the MQINI resource cannot be found.

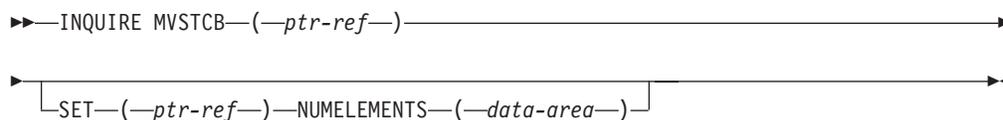
NOTAUTH

RESP2 values:

- 100 Command authorization failure.

INQUIRE MVSTCB

Retrieve addresses and storage usage information for MVS TCBs.

INQUIRE MVSTCB

Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.

Description

The INQUIRE MVSTCB command can only be used in browse mode. It returns addresses and storage information for the MVS TCBs in the CICS address space. The information for each TCB shows the addresses, lengths and MVS subpools for the storage elements owned by the TCB, the storage key for each element, and the number of bytes in use (GETMAINED by the task) for each element.

The correct syntax for this command for all new applications is shown above. The options ELEMENTLIST, LENGTHLIST and SUBPOOLLIST in the listing of options below are obsolete, but are supported for compatibility with applications developed in releases before CICS Transaction Server for z/OS, Version 3 Release 2. You cannot use these options in combination with the SET option.

The NUMELEMENTS option has a role in both the old syntax and the new syntax. Where the ELEMENTLIST, LENGTHLIST and SUBPOOLLIST options are used, the NUMELEMENTS option specifies the number of entries in each of these lists (which is the same for each list). NUMELEMENTS is also used in combination with the SET option, to give the number of addresses in the pointer list returned by the SET option.

Browsing

This command can only be used in browse mode. Browse through all of the MVS TCBs in the CICS address space by using the browse options (START, NEXT, and END) on the command. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

Options

ELEMENTLIST(*ptr-ref*)

returns the address of a list of the addresses of all areas of private storage allocated to this TCB. This option is obsolete, but it is supported for compatibility with applications developed in earlier CICS releases.

LENGTHLIST(*ptr-ref*)

returns the address of a list of fullword binary lengths of the storage areas listed in the ELEMENTLIST list. This option is obsolete, but it is supported for compatibility with applications developed in earlier CICS releases.

NUMELEMENTS(*data-area*)

A fullword binary field which is set to the number of storage elements owned by this TCB. This value is the number of addresses listed in the pointer list returned by the SET option, where each address indicates one storage element.

MVSTCB(*ptr-ref*)

returns the address of the MVS TCB in the CICS address space. The TCB address that is returned can be used as input to the EXEC CICS COLLECT STATISTICS MVSTCB command to retrieve storage and CPU time statistics for the TCB.

SET(*ptr-ref*)

returns the address of a list of four-byte addresses. Each address points to a descriptor containing details of one storage element owned by this TCB. The number of addresses in the list is the value returned by the NUMELEMENTS option.

CICS obtains the storage for the list and descriptors. It is freed when the inquiring task ends or issues another INQUIRE MVSTCB command with one of the command options. The task cannot free the storage itself.

The format of the descriptor for each storage element is shown in Table 3:

Table 3. INQUIRE MVSTCB, SET option: Descriptor for each storage element

Offset (decimal)	Length	Contents
0	4	Address of the storage
4	4	Length
8	4	MVS subpool number
12	4	MVS storage key (for example, 8)
16	4	Number of bytes in use

Note: “Number of bytes in use” is the amount of storage GETMAINED by the task. This might be less than the amount of storage allocated to the TCB, because storage is always allocated to a TCB in page multiples (4096 bytes).

SUBPOOLLIST(*ptr-ref*)

returns the address of a list of fullword binary subpool numbers of the MVS

subpools for the storage areas listed in the ELEMENTLIST list. This option is obsolete, but it is supported for compatibility with applications developed in earlier CICS releases.

Conditions

END

RESP2 values:

- 2 All authorized resources have been retrieved. All data areas specified on this command are left unchanged.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

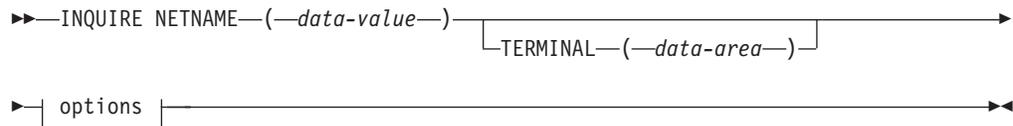
RESP2 values:

- 1 The TCB specified on the command was not found.

INQUIRE NETNAME

Retrieve information about a terminal or session.

INQUIRE NETNAME/TERMINAL



options:

— for options, see INQUIRE TERMINAL —

Conditions: NOTAUTH, TERMIDERR

Description

The INQUIRE NETNAME command returns information about a particular terminal or session, just as the INQUIRE TERMINAL command does. The primary difference is that you identify the terminal by its network identifier, instead of its CICS terminal identifier. For a physical terminal, the network identifier is the name by which the terminal is known to z/OS Communications Server. For ISC sessions, it is the name by which the session (or session group, if there are parallel sessions) is known to z/OS Communications Server. For MRO sessions, it is the name used by the connected region to log on to the interregion communication program.

Thus, compared with INQUIRE TERMINAL, the roles of the NETNAME and TERMINAL options are reversed; NETNAME is required, and you supply a data-value containing the 8-character network identifier of the terminal about which you are inquiring. TERMINAL is optional. If you use it, CICS returns the corresponding 4-character CICS terminal identifier in the data-area you provide.

The other options for INQUIRE TERMINAL return the same information in an INQUIRE NETNAME command as they do in an INQUIRE TERMINAL command.

If there are multiple entries for a netname, and the inquiry is not part of a browse, the first entry found is returned. Entries are searched in the following sequence:

1. z/OS Communications Server terminals and consoles, in alphanumeric sequence.
2. Connections, in alphanumeric sequence. The leading session is returned (in a browse, all sessions are returned).

Note that this order is not guaranteed to be maintained in future releases.

Browsing

You can also browse through the definitions of all the netnames installed in your system by using the browse options (START, NEXT, and END) on INQUIRE NETNAME or INQUIRE TERMINAL commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

Note that connections without sessions, such as indirect connections, or remote connections that are not in use, are not returned. If you want all connections to be returned, use the INQUIRE CONNECTION command.

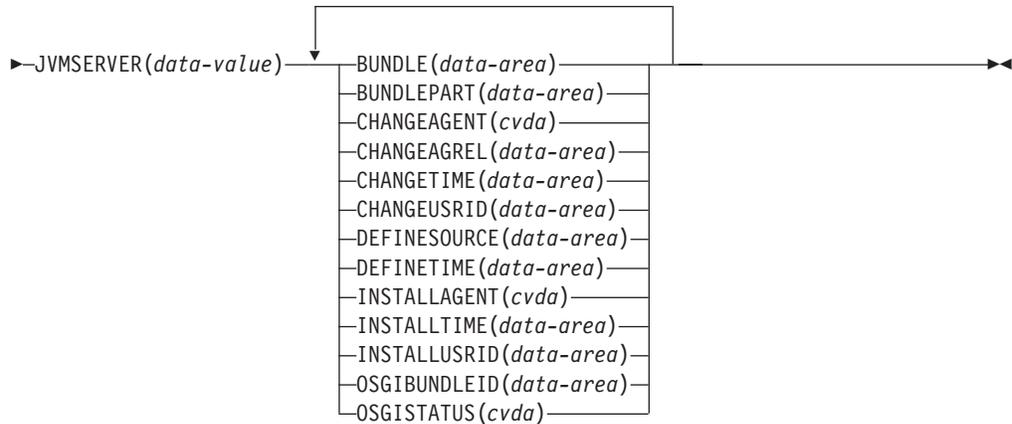
See “INQUIRE TERMINAL” on page 485 for details of the options and conditions that apply to the INQUIRE NETNAME command.

INQUIRE OSGIBUNDLE

Retrieve information about an OSGi bundle that is installed in a JVM server.

INQUIRE OSGIBUNDLE

►►—INQUIRE OSGIBUNDLE(*data-value*)—OSGIVERSION(*data-value*)—————►



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.

For more information about the use of CVDAs, see CICS-value data areas (CVDA) in CICS System Programming Reference.

Description

Use the **INQUIRE OSGIBUNDLE** command to find information about an OSGi bundle that is installed in a JVM server.

Browsing

You can browse through all the OSGi bundles that are installed in a particular JVM server by using the browse options, START, NEXT, and END, on **INQUIRE OSGIBUNDLE** commands. When browsing, specify the JVMSEVER resource that you want to browse. See Browsing resource definitions in CICS System Programming Reference for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

BUNDLE(*data-area*)

Returns the 8-character name of the CICS BUNDLE resource that contains the deployed OSGi bundle on the specified JVM server.

| **BUNDLEPART**(*data-area*)

| Returns the 255-character name of the part of the CICS BUNDLE resource that
| represents the installed OSGi bundle in the specified JVM server.

| **CHANGEAGENT**(*cvda*)

| Returns a CVDA value that identifies the agent that made the last change to
| the resource definition. The possible values are as follows:

| **CREATESPI**

| The resource definition was last changed by an **EXEC CICS CREATE**
| command.

| **CSDAPI**

| The resource definition was last changed by a CEDA transaction or the
| programmable interface to DFHEDAP.

| **CSDBATCH**

| The resource definition was last changed by a DFHCSDUP job.

| **DREPAPI**

| The resource definition was last changed by a CICSplex SM BAS API
| command.

| **CHANGEAGREL**(*data-area*)

| Returns a 4-digit number of the CICS release that was running when the
| resource definition was last changed.

| **CHANGETIME**(*data-area*)

| Returns an ABSTIME value that represents the time stamp when the resource
| definition was last changed. For more information about the format of the
| ABSTIME value, see FORMATTIME in CICS Application Programming.

| **CHANGEUSRID**(*data-area*)

| Returns the 8-character user ID that ran the change agent.

| **DEFINESOURCE**(*data-area*)

| Returns the 8-character source of the resource definition. The DEFINESOURCE
| value depends on the CHANGEAGENT value. For more information, see
| Summary of the resource signature field values in the Resource Definition
| Guide.

| **DEFINETIME**(*data-area*)

| Returns an ABSTIME value that represents the time stamp when the resource
| definition was created.

| **JVMSERVER**(*data-value*)

| Specifies the 8-character name of the JVMSERVER resource in which the OSGi
| bundle is installed.

| **INSTALLAGENT**(*cvda*)

| Returns a CVDA value that identifies the agent that installed the resource.
| Only one value is possible:

| **BUNDLE**

| The resource was installed by a bundle deployment.

| **INSTALLTIME**(*data-area*)

| Returns an ABSTIME value that represents the time stamp when the resource
| was installed.

| **INSTALLUSRID**(*data-area*)

| Returns the 8-character user ID that installed the resource.

| **OSGIBUNDLE**(*data-value*)

| Specifies the 255-character symbolic name of the OSGi bundle.

| **OSGIBUNDLEID**(*data-area*)

| Returns a doubleword binary value of the bundle ID in the OSGi framework.

| **OSGISTATUS**(*cvda*)

| Returns the status of the OSGi bundle. The status can be one of the following values:

| **ACTIVE**

| The bundle has been successfully activated and is running; its bundle activator start method has been called and returned.

| **INSTALLED**

| The OSGi bundle has successfully installed.

| **RESOLVED**

| All Java classes that the bundle requires are available. This state indicates that the bundle is either ready to be started or has stopped.

| **STARTING**

| The bundle is being started in the OSGi framework. The bundle activator start method is called but has not yet returned. When the bundle has an activation policy, the bundle remains in the STARTING state until the bundle is activated according to its activation policy.

| **STOPPING**

| The bundle is being stopped in the OSGi framework. The bundle activator stop method is called but has not yet returned.

| **UNINSTALLED**

| The bundle has been uninstalled. It cannot move into another state.

| **OSGIVERSION**(*data-value*)

| Specifies the 255-character symbolic version of the OSGi bundle.

| **Conditions**

| **END**

| RESP2 values:

| 2 There are no more resource definitions of this type.

| **ILLOGIC**

| RESP2 values:

| 1 You have either issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

| **NOTAUTH**

| RESP2 values:

| 100 The user associated with the issuing task is not authorized to use this command.

| 101 The user associated with the issuing task is not authorized to access this bundle.

| **NOTFND**

| RESP2 values:

| 1 The JVMSERVER resource is not found or disabled.

INQUIRE OSGISERVICE

Retrieve information about OSGi services that are registered in a CICS region.

INQUIRE OSGISERVICE

►►—INQUIRE OSGISERVICE(*data-value*)—JVMSERVER(*data-value*)—►►



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.

For more information about the use of CVDAs, see CICS-value data areas (CVDA) in CICS System Programming Reference.

Description

Use the **INQUIRE OSGISERVICE** command to find information about an OSGi service that is registered in a JVM server.

Browsing

You can browse through all the OSGi services that are registered in a JVM server by using the browse options, START, NEXT, and END, on **INQUIRE OSGISERVICE** commands. When browsing, you must specify the name of the JVMSERVER resource. See Browsing resource definitions in CICS System Programming Reference for general information about browsing, including syntax, exception conditions, and examples.

Options

BUNDLE(*data-area*)

Returns the 8-character name of the CICS BUNDLE resource that contains the OSGi service on the specified JVM server.

BUNDLEPART(*data-area*)

Returns the 255-character name of the part of the CICS BUNDLE resource that represents the installed OSGi bundle in the specified JVM server.

JVMSERVER(*data-value*)

Specifies the 8-character name of the JVMSERVER resource in which the OSGi service is registered.

| **OSGIBUNDLE**(*data-area*)

| Returns the 255-character symbolic name of the OSGi bundle that contains the
| OSGi service.

| **OSGISERVICE**(*data-value*)

| Specifies a doubleword binary value representing the ID of the OSGi service.

| **OSGIVERSION**(*data-area*)

| Returns the 255-character version of the OSGi bundle that defines the OSGi
| service.

| **SRVCNAME**(*data-area*)

| Returns the 255-character name of the OSGi service that is registered in the
| JVM server.

| **SRVCSTATUS**(*cvda*)

| Returns the status of the OSGi service. The status can have one of the
| following values:

| **ACTIVE**

| The OSGi service is available in the OSGi framework.

| **INACTIVE**

| The OSGi service is not available in the OSGi framework, typically
| because it is a duplicate of another active OSGi service.

| **Conditions**

| **END**

| RESP2 values:

| **2** There are no more resource definitions of this type.

| **ILLOGIC**

| RESP2 values:

| **1** You have either issued a START command when a browse of this
| resource type is already in progress, or you have issued a NEXT or an
| END command when a browse of this resource type is not in progress.

| **NOTAUTH**

| RESP2 values:

| **100** The user associated with the issuing task is not authorized to use this
| command.

| **101** The user associated with the issuing task is not authorized to access
| this bundle.

| **NOTFND**

| RESP2 values:

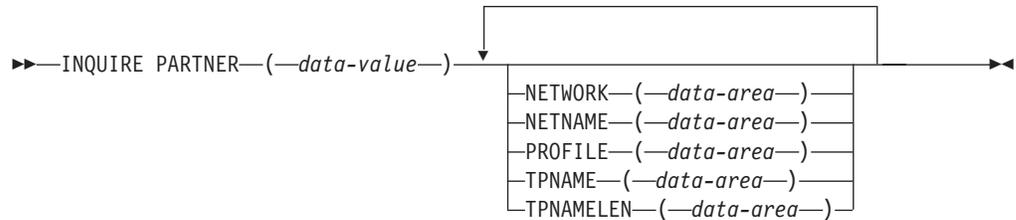
| **1** The JVMSERVER resource cannot be found.

| **3** The OSGi service cannot be found.

INQUIRE PARTNER

Retrieve information about a partner.

INQUIRE PARTNER



Conditions: END, ILLOGIC, NOTAUTH, PARTNERIDERR

Description

The INQUIRE PARTNER command returns information about a partner from the partner resource table.

Browsing

You can also browse through all of the partners defined in your system by using the browse options (START, NEXT, and END) on INQUIRE PARTNER commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

Options

NETNAME(*data-area*)

returns the 8-character name of the z/OS Communications Server node in which the partner is located.

NETWORK(*data-area*)

returns the 8-character name of the network in which the partner is located. If this value is blank, the partner is in the same network as your CICS system.

PARTNER(*data-value*)

specifies the 8-character name of the partner about which you are inquiring. This is the name assigned in its PARTNER resource definition.

PROFILE(*data-area*)

returns the 8-character name of the PROFILE definition specified in the PARTNER definition.

TPNAME(*data-area*)

returns the name of the remote transaction program that runs on the partner LU (from the TPNAME or XTPNAME value in the PARTNER resource definition). This name can be up to 64 characters long; you can determine the actual length with the TPNAMELEN option.

TPNAMELEN(*data-area*)

returns a halfword binary field giving the length in bytes of the information returned in TPNAME.

Conditions

END

RESP2 values:

2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

PARTNERIDERR

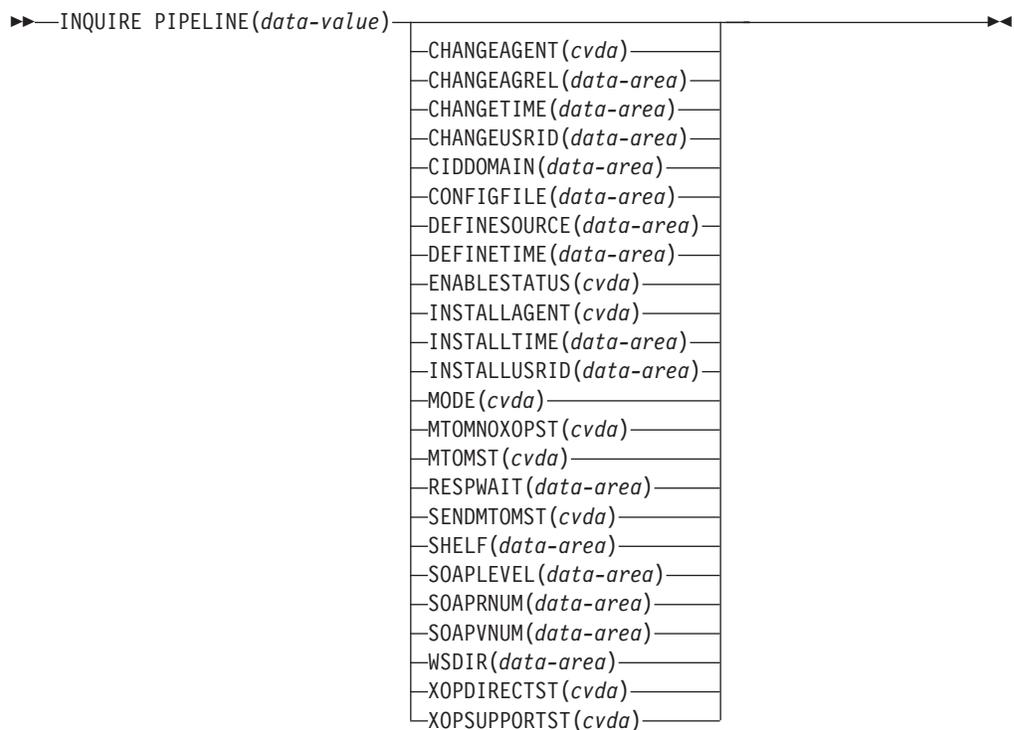
RESP2 values:

- 1 The partner cannot be found.
- 2 Partner Resource Manager (PRM) is not active, because it failed to initialize during CICS initialization.

INQUIRE PIPELINE

Retrieve information about an installed PIPELINE.

INQUIRE PIPELINE



Conditions: NOTAUTH, NOTFND

This command is threadsafe.

For more information about the use of CVDAs, see CICS-value data areas (CVDA) in CICS System Programming Reference.

Description

Use the **INQUIRE PIPELINE** command to retrieve information about an installed pipeline.

Browsing

You can browse through all the pipelines installed in your system by using the browse options, **START**, **NEXT**, and **END**, on **INQUIRE PIPELINE** commands. See Browsing resource definitions in CICS System Programming Reference for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are **CHANGEAGENT**, **CHANGEAGREL**, **CHANGETIME**, **CHANGEUSRID**, **DEFINESOURCE**, **DEFINETIME**, **INSTALLAGENT**, **INSTALLTIME**, and **INSTALLUSRID**. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

PIPELINE (*data-value*)

Specifies the name of the pipeline about which you are inquiring. The name can be up to 8 characters long.

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

CIDDOMAIN(*data-area*)

Returns the domain name that is used to generate MIME content-ID values to identify binary attachments in containers. The name can be up to 255 characters long.

CONFIGFILE(*data-area*)

Returns the name of the pipeline configuration file associated with the pipeline resource. The name can be up to 255 characters long.

DEFINESOURCE(*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

ENABLESTATUS(*cvda*)

Returns the status of the PIPELINE:

ENABLED

The pipeline is ready for use.

DISABLED

The pipeline is not processing requests, and cannot to accept new work. It might have failed to initialize or might have been explicitly disabled.

ENABLING

The pipeline is being initialized; it is not yet ready to accept work.

DISABLING

The pipeline is quiescing before entering DISABLED state. It is not accepting new work, but is allowing current work to complete.

DISCARDING

A DISCARD command has been issued for the pipeline. The pipeline is quiescing before being discarded. It is not accepting new work, but is allowing current work to complete.

INSTALLAGENT(*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

MODE(*cvda*)

Returns the operating mode of the pipeline. CVDA values are as follows:

PROVIDER

CICS is using the pipeline as a service provider.

REQUESTER

CICS is using the pipeline as a service requester.

UNKNOWN

The operating mode of the pipeline cannot be determined.

MTOMNOXOPST (*cvda*)

Returns a value that indicates whether MTOM will be used for outbound SOAP messages when no binary attachments are present.

MTOMNOXOP

Use MTOM, even when no binary attachments are present.

NOMTOMNOXOP

Do not use MTOM unless binary attachments are present.

MTOMST (*cvda*)

Returns a value that indicates whether support for MTOM has been enabled in the pipeline.

MTOM

MTOM support has been enabled in the pipeline.

NOMTOM

MTOM support has not been enabled in the pipeline.

RESPWAIT (*data-area*)

Returns the number of seconds that an application program waits for an optional response message from a remote Web service. If the returned value is -1, no value has been set for the pipeline and the default timeout value of the transport protocol is being used.

- The default timeout value for HTTP is 10 seconds.
- The default timeout value for WebSphere MQ is 60 seconds.

SENDMTOMST (*cvda*)

Returns a value that indicates when MTOM will be used for outbound SOAP messages.

NOSENDMTOM

Do not use MTOM for outbound SOAP messages.

SAMESENDMTOM

Use MTOM for outbound SOAP message responses when the inbound message is received in MTOM format.

SENDMTOM

Always use MTOM for outbound SOAP messages.

SOAPLEVEL (*data-area*)

Returns an 8-byte character string of the SOAP level that is used in the PIPELINE. The value of the SOAP level is 1.1 or 1.2. If the pipeline is not being used for SOAP messages, a value of NOTSOAP is returned.

SOAPRNUM (*data-area*)

Returns a fullword binary value of the release number for the SOAP level that is used in the PIPELINE. The value of the release number is 1 or 2.

SOAPVNUM (*data-area*)

Returns a fullword binary value of the version number for the SOAP level that is used in the PIPELINE. The value of the version number is 1.

SHELF(*data-area*)

Returns the name of the shelf directory. The name can be up to 255 characters long.

WSDIR(*data-area*)

Returns the name of the Web service binding directory (also known as the pickup directory). The name can be up to 255 characters long.

XOPDIRECTST(*cvda*)

Returns a value that indicates whether the pipeline can currently handle XOP documents in direct mode.

XOPDIRECT

The pipeline supports the direct processing of XOP documents and binary attachments.

NOXOPDIRECT

The pipeline does not support the direct processing of XOP documents and binary attachments. Compatibility mode is in operation.

XOPSUPPORTST(*cvda*)

Returns a value that indicates whether the application handler for the pipeline supports the processing of XOP documents and binary attachments.

XOPSUPPORT

The application handler supports XOP documents.

NOXOPSUPPORT

The application handler does not support XOP documents.

Conditions**NOTAUTH**

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

NOTFND

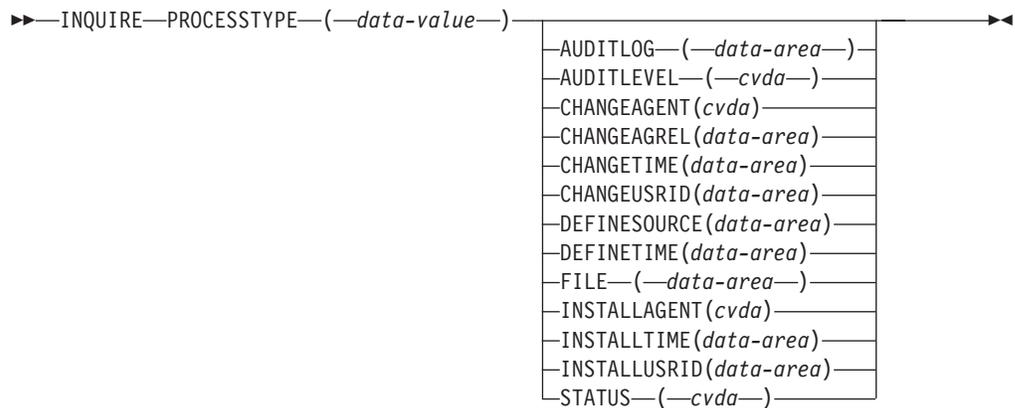
RESP2 values:

3 The PIPELINE cannot be found.

INQUIRE PROCESSTYPE

Retrieve the attributes of a CICS business transaction services (CBTS) process type.

INQUIRE PROCESSTYPE



Conditions: NOTAUTH, PROCESSERR

Description

INQUIRE PROCESSTYPE returns the attributes of a specified process type.

Browsing

You can also browse through all of the process type definitions in your system by using the browse options, START, NEXT, and END, on INQUIRE PROCESSTYPE commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

AUDITLEVEL(*cvda*)

Indicates the level of audit currently active for processes of the specified type. CVDA values are as follows:

ACTIVITY

Activity-level auditing. Audit records are written from these points:

- The process audit points
- The activity primary audit points.

FULL Full auditing. Audit records are written from these points:

- The process audit points
- The activity primary *and* secondary audit points.

OFF No audit trail records are written.

PROCESS

Process-level auditing. Audit records are written from the process audit points only.

For details of the records that are written from the process, activity primary, and activity secondary audit points, see *CICS Business Transaction Services*.

AUDITLOG(*data-area*)

Returns the 8-character name of the CICS journal used as the audit log for processes of the specified type.

CHANGEAGENT(*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL(*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID(*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE(*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT** value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

FILE(*data-area*)

Returns the 8-character name of the CICS file associated with the process-type.

INSTALLAGENT(*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

PROCESSTYPE(*data-value*)

Specifies the name (1 - 8 characters) of the process type being inquired on.

STATUS(*cvda*)

Indicates whether new processes of the specified type can currently be defined. CVDA values are as follows:

DISABLED

The installed definition of the process type is disabled. New processes of this type cannot be defined.

ENABLED

The installed definition of the process type is enabled. New processes of this type can be defined.

Conditions**NOTAUTH**

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

101 The user associated with the issuing task is not authorized to access this resource in the way requested.

PROCESSERR

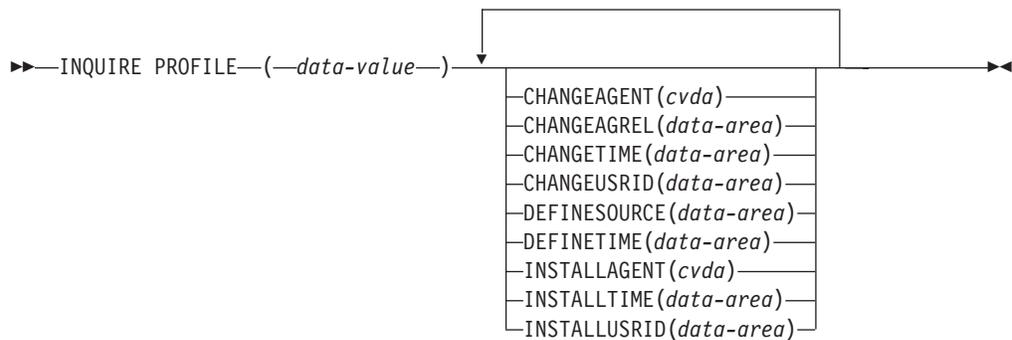
RESP2 values:

1 The process type specified on the PROCESSTYPE option cannot be found.

INQUIRE PROFILE

Determine whether a transaction profile is installed.

INQUIRE PROFILE



Conditions: END, ILLOGIC, NOTAUTH, PROFILEIDERR

Description

Use the INQUIRE PROFILE command to determine whether a particular PROFILE definition is installed in your CICS system. The command has no options; you get a normal response if the profile about which you inquire is installed in your CICS system and a PROFILEIDERR exception condition if it is not.

Browsing

You can also use the INQUIRE PROFILE command in browse form (the START, NEXT, and END options) to obtain the names of all of the profiles installed in your system. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT (cvda)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

SYSTEM

The resource definition was last changed by the CICS or CICSplex system.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see FORMATTIME in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID (*data-area*)

Returns the 8-character user ID that installed the resource.

PROFILE (*data-value*)

Specifies the 8-character name of the profile about which you are inquiring.

Conditions**END**

RESP2 values:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

PROFILEIDERR

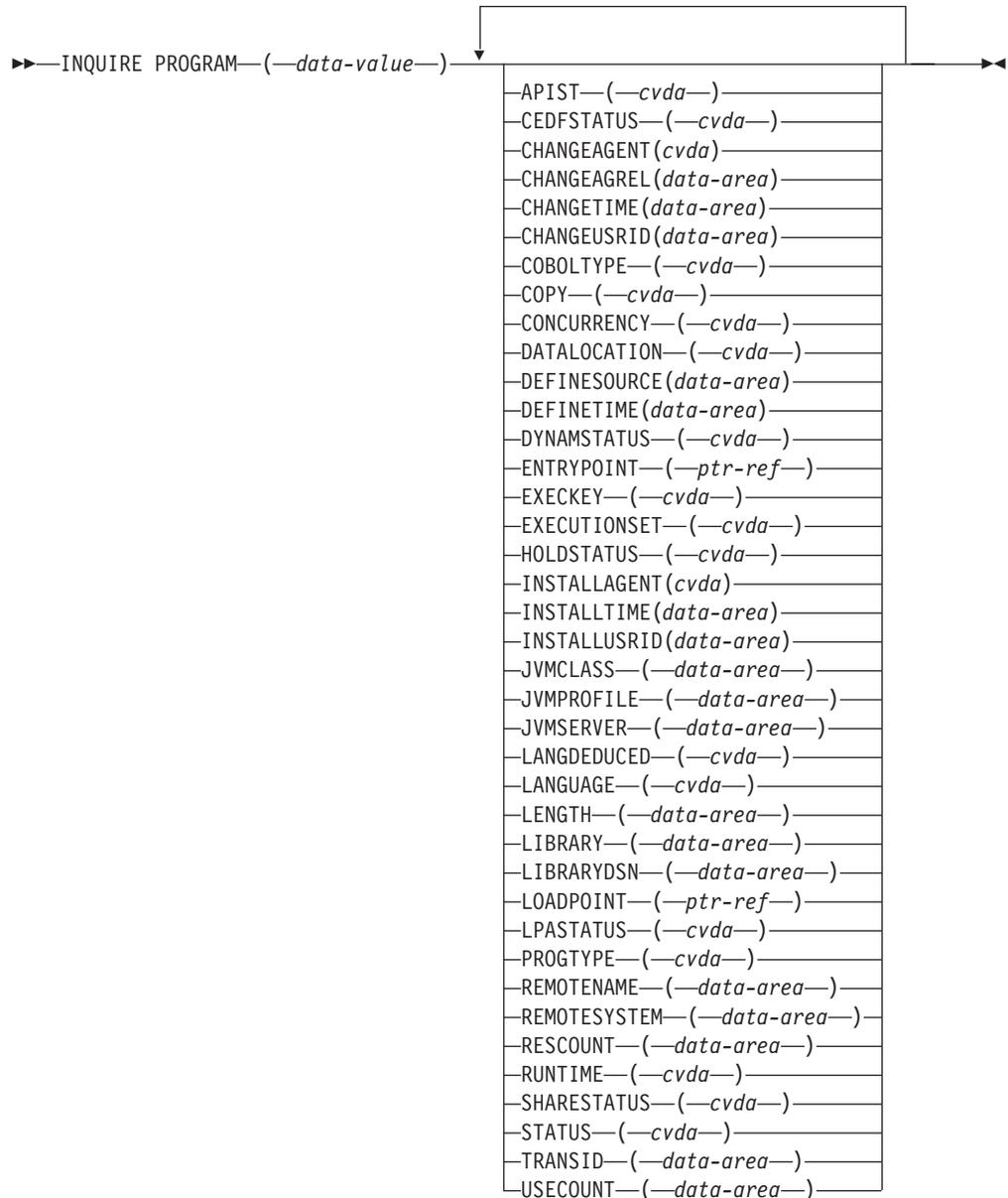
RESP2 values:

- 1 The profile cannot be found.

INQUIRE PROGRAM

Retrieve information about a program, map set, or partition set.

INQUIRE PROGRAM



Conditions: END, ILLOGIC, NOTAUTH, PGMIDERR

For more information about the use of CVDAs, see CICS-value data areas (CVDAs) in CICS System Programming Reference.

Description

The **INQUIRE PROGRAM** command returns information about a particular program, map set, or partition set installed in your CICS region. All of these resources are load modules and, therefore, CICS uses the same INQUIRE command for all three. To avoid confusion, the word *module* refers to the object of your inquiry, except in some cases where the option applies only to executable programs.

CICS determines the information that you request from both the resource definition and, where applicable, the load module. Information from the module

takes precedence over that in the definition if there is a conflict. However, CICS inspects a module only if it is already loaded and is the copy currently available for use. CICS does not do a load for an INQUIRE PROGRAM command, nor attempt to autoinstall a resource for which it has no definition.

Browsing

You can also browse through the definitions of these three types of resources in your system by using the browse options, START, AT, NEXT, and END, on INQUIRE PROGRAM commands. In browse mode, the definitions are returned in alphabetic order, and you can specify a starting point with the AT option if you want. See Browsing resource definitions in CICS System Programming Reference for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

APIST(*cvda*) (programs only)

Returns a CVDA value indicating the API attribute of the installed program definition. The API attribute is used for application programs, PLT programs, user-replaceable modules, and task-related user exits. The API attribute is not used for global user exits.

The API attribute on the installed PROGRAM resource for a task-related user exit program is not changed by any options specified on an ENABLE command. For a task-related user exit program, CICS always returns a CVDA using the values defined in the program resource definition.

You cannot modify an API attribute of a program using the SPI. You can change the API attribute only by redefining the API option of the program in the CICS PROGRAM resource definition, or in the program autoinstall model, and reinstalling the definition. CVDA values are as follows:

CICSAPI

The program is restricted to use of the CICS permitted application programming interfaces only. If the program is defined with CONCURRENCY(QUASIRENT) it always runs on the quasi-reentrant (QR) TCB, if it is defined as CONCURRENCY(THREADSAFE) it runs on whichever TCB in use by CICS at the time that is determined as suitable, and if it is defined as CONCURRENCY(REQUIRED) it always runs on an L8 open TCB.

OPENAPI

The program is not restricted to the CICS permitted application programming interfaces only. CICS runs the program on its own L8 or L9 mode open TCB depending on the EXECKEY setting. If, when

running a CICS command, CICS requires a switch to QR TCB, it returns to the open TCB before handing control back to the application program.

OPENAPI requires that the program is coded to threadsafe standards and defined with CONCURRENCY(THREADSAFE) or CONCURRENCY(REQUIRED). The preferred option is to use CONCURRENCY(REQUIRED) with OPENAPI, although CONCURRENCY(THREADSAFE) is allowed for compatibility with previous releases.

CEDFSTATUS(*cvda*) (programs only)

Returns a CVDA value indicating the action taken by the execution diagnostic facility (EDF) transaction if this module is executed under EDF. CVDA values are as follows:

CEDF EDF diagnostic screens are displayed. If the program was translated with the EDF option, all EDF screens are displayed; if it was translated with NOEDF, only the program initiation and termination screens appear.

NOCEDF

No EDF screens are displayed.

NOTAPPLIC

EDF does not apply because the module is a remote program, a map set, or a partition set.

CHANGEAGENT(*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

AUTOINSTALL

The resource was autoinstalled.

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

SYSTEM

The resource definition was last changed by the CICS or CICSplex system.

CHANGEAGREL(*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID(*data-area*)

Returns the 8-character user ID that ran the change agent.

COBOLTYPE(*cvda*) (**programs only**)

Returns a CVDA value indicating the type of COBOL in which the module is written, if it is a COBOL program. The type is determined by inspecting the load module. CVDA values are as follows:

COBOL

The module is an OS/VS COBOL program. (OS/VS COBOL programs cannot run under this CICS Transaction Server version.)

COBOLII

The module is a COBOL program that has been compiled with VS COBOL II or a more recent COBOL compiler.

NOTAPPLIC

The module has been loaded and it is not a COBOL program, or the module has not been loaded and it is not defined as a COBOL program.

NOTINIT

The module is defined as a COBOL program, but the type cannot be determined because the module has not been loaded yet.

CONCURRENCY

Returns a CVDA value indicating the concurrency attribute of the installed program definition. The CVDA values are as follows:

QUASIRENT

The program is defined as being quasi-reentrant, and can run only under the CICS QR TCB.

THREADSAFE

The program is defined as threadsafe, and can run under whichever TCB is in use by its user task when the program is given control. This can be either an open TCB or the CICS QR TCB.

REQUIRED

The program is defined as threadsafe, and must run on an open TCB. The type of open TCB used depends on the API setting.

Notes:

1. If the program is not yet loaded, or is waiting to be reloaded following a NEWCOPY or PHASEIN request, the concurrency attribute is derived from the installed program resource definition. Note that the default for the program definition is QUASIRENT. However, in the case of a Language Environment-conforming program, the concurrency as originally defined can be overridden when the program is later loaded. If CICS finds that the program itself contains a CONCURRENCY value defined by Language Environment[®] runtime options, the installed program resource definition is updated by the Language Environment runtime option.
2. The CONCURRENCY attribute on the installed program resource definition is not changed by the FORCEQR system initialization parameter. CICS returns a CVDA of THREADSAFE for a threadsafe-defined program, even if FORCEQR=YES is specified.
3. The CONCURRENCY attribute on the installed program resource definition for a task-related user exit program is not changed by any options specified

on an ENABLE command. For a task-related user exit program, CICS always returns a CVDA using the values defined in the program resource definition.

You cannot modify the concurrency attribute of a program using the SPI; the CONCURRENCY option is not supported on the EXEC CICS SET PROGRAM command. You can change the concurrency only by redefining the CONCURRENCY option of the program in the CICS program resource definition, or in the program autoinstall model, and then reinstalling the definition.

COPY(*cvda*)

Returns a CVDA value indicating whether a new copy of the module is required to make it available for use. This requirement occurs after CICS attempts to load the module and cannot find it, because CICS marks it “not loadable” to avoid the overhead of further load attempts. To make the module available again, issue a SET PROGRAM COPY command or its CEMT equivalent. Ensure that the program exists in one of the libraries in the DFHRPL or dynamic LIBRARY concatenation before doing so. CVDA values are as follows:

NOTREQUIRED

A new copy is not required. This CVDA value is always returned for Java programs that run in a JVM.

REQUIRED

A new copy is required.

DATALOCATION(*cvda*) (programs only)

Returns a CVDA value indicating whether this module can accept data addresses higher than 16MB. CVDA values are as follows:

ANY The program can accept an address above 16 MB.

BELOW

The program requires any data address returned to it from CICS to be less than 16 MB.

NOTAPPLIC

The option does not apply because the module is a remote program, a map set, or a partition set.

DEFINESOURCE(*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

DYNAMSTATUS(*cvda*) (programs only)

Returns a CVDA value indicating whether, if the program is the subject of a program-link request, the request can be dynamically routed. CVDA values are as follows:

DYNAMIC

If the program is the subject of a program-link request, the CICS dynamic routing program is invoked. Providing that a remote server

region is not named explicitly on the SYSID option of the LINK command, the routing program can route the request to the region on which the program is to run.

NOTDYNAMIC

If the program is the subject of a program-link request, the dynamic routing program is not invoked.

For a distributed program link (DPL) request, the server region on which the program is to run must be specified explicitly on the REMOTESYSTEM option of the PROGRAM definition or on the SYSID option of the LINK command; otherwise, it defaults to the local region.

For information about the dynamic routing of DPL requests, see the *CICS Intercommunication Guide*.

ENTRYPOINT (*ptr-ref*)

Returns the entry point of the module, if it is loaded. The top bit of the address is set on if the addressing mode is 31 and off if it is 24. If the module has not been loaded, or is a remote program, or is a Java program that runs in a JVM, a null pointer (X'FF000000') is returned.

EXECKEY (*cvda*) (**programs only**)

Returns a CVDA value indicating the storage key of the module, if it is an executable program. The storage key can limit the areas of storage that the program can access, depending on other variables. See the ISOLATEST option of the INQUIRE TASK and INQUIRE TRANSACTION commands, the STOREPROTECT and TRANISOLATE options of the INQUIRE SYSTEM command, and the general discussion of storage protection in the *CICS Application Programming Guide*. CVDA values are as follows:

CICSEXECKEY

The program runs in CICS key.

NOTAPPLIC

The module is a remote program, a map set, or a partition set.

USEREXECKEY

The program runs in user key.

EXECUTIONSET (*cvda*) (**programs only**)

Returns a CVDA value indicating whether the module is restricted to the distributed program link subset of the CICS API. The EXECUTIONSET option applies only to executable programs, and governs the API only when a program is called locally. When it is called remotely, that is, executing at or below the level of a program invoked by a distributed program link, a program is always restricted to this subset. CVDA values are as follows:

DPLSUBSET

The program is always restricted.

FULLAPI

The program is not restricted unless called remotely.

NOTAPPLIC

EXECUTIONSET does not apply because the module is a remote program, a map set, or a partition set.

HOLDSTATUS (*cvda*)

Returns a CVDA value indicating whether a copy of the module is currently loaded with the HOLD option. CVDA values are as follows:

HOLD

A copy is currently loaded with the HOLD option.

NOHOLD

No copy is currently loaded with the HOLD option.

NOTAPPLIC

The module is not currently loaded, or is a remote program.

INSTALLAGENT(*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

AUTOINSTALL

The resource was autoinstalled.

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

SYSTEM

The resource was installed by the CICS or CICSplex SM system.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

JVMCLASS(*data-area*) (Java programs only)

Returns the name, in 255 characters, of the main class in the Java program to be given control by the JVM, as specified in the program definition.

JVMPROFILE(*data-area*) (Java programs only)

Returns the name of the JVM profile that is to be used for the pooled JVM in which this Java program runs. The name can be up to 8 characters in length.

JVMSERVER(*data-area*) (Java programs only)

Returns the name of the JVM server in which this Java program runs. The name can be up to 8 characters in length.

LANGDEDUCED(*cvda*) (programs only)

Returns a CVDA value indicating the language for the loaded module. If the module is not yet loaded, CICS cannot deduce the language. In this case, the CVDA value indicates the defined language taken from the resource definition. CVDA values are as follows:

ASSEMBLER

The language is assembler.

C

The language is C or C++.

COBOL

The language is COBOL.

JAVA

The language is Java.

LE370 The module, whatever its language, was compiled to run with Language Environment.

NOTAPPLIC

LANGUAGE does not apply because the module is a remote program, a map set, or a partition set.

NOTDEFINED

The language was not specified in the resource definition, and has not been loaded.

PLI or PL1

The language is PL/I.

LANGUAGE(*cvda*) (programs only)

Returns a CVDA value indicating the program language. The CICS program manager deduces the correct language, except for programs written in assembler without the DFHEAI stub. In this case, the LANGUAGE attribute of the program definition is used to return a value. CVDA values are as follows:

ASSEMBLER

The language is assembler.

C The language is C.

COBOL

The language is COBOL

LE370 The module, whatever its language, exploits multi-language support, or was compiled with a Language Environment-conforming compiler.

NOTAPPLIC

LANGUAGE does not apply because the module is a remote program, a map set, or a partition set.

NOTDEFINED

The language was not specified in the resource definition.

PLI or PL1

The language is PL/I.

LENGTH(*data-area*)

Returns a fullword binary field giving the length of the module in bytes. A value of 0 is returned if the module has not been loaded in the current CICS session. A value of -1 is returned if it is a remote program or a Java program that runs in a JVM.

LIBRARY(*data-area*)

Returns the 8-character name of the library resource from which this program was loaded. This data area is blank if the program has not been loaded, or if the LPASTATUS is LPA, indicating that the program has been loaded from the LPA.

Note:

- If the program was loaded from an installed library, the LIBRARY and LIBRARYDSN names are returned.
- If the program was loaded from a library that has been disabled, the LIBRARY name is returned but the LIBRARYDSN is blank.
- If the program was loaded from a library that has been discarded, both LIBRARY and LIBRARYDSN names are blank.

LIBRARYDSN (*data-area*)

Returns the 44-character name of the data set from which the program was loaded. This data area is blank if the program has not been loaded, or if the LPASTATUS is LPA, indicating that the program has been loaded from the LPA.

- If the program was loaded from an installed library, the LIBRARY and LIBRARYDSN names are returned.
- If the program was loaded from a library that has been disabled, the LIBRARY name is returned but the LIBRARYDSN is blank.
- If the program was loaded from a library that has been discarded, both LIBRARY and LIBRARYDSN names are blank.

LOADPOINT (*ptr-ref*)

Returns the load address of the module. If it is not currently loaded, or if the program is a Java program running in a JVM, a null pointer (X'FF000000') is returned.

LPASTATUS (*cvda*)

Returns a CVDA value indicating whether the module resided in the link pack area when it was last used. CVDA values are as follows:

LPA The copy used was in the link pack area (LPA) or the extended link pack area (ELPA).

NOTAPPLIC

The module has not been used, is a remote program, or is a Java program.

NOTLPA

The copy used was in CICS dynamic storage.

PROGRAM (*data-value*)

Specifies the name of the program, map set, or partition set about which you are inquiring. The name can be up to 8 characters in length.

PROGTYPE (*cvda*)

Returns a CVDA value indicating the type of module. CVDA values are as follows:

MAPSET

The module is a map set. (MAP is still a synonym for MAPSET, but MAPSET is the preferred CVDA value.)

PARTITIONSET

The module is a partition set.

PROGRAM

The module is an executable program.

REMOTENAME (*data-area*) (**programs only**)

Returns the 8-character name by which the module is known in the CICS region named in the REMOTESYSTEM option of its PROGRAM definition. The REMOTENAME option applies only to programs, and only to those defined to be remote; for local programs, map sets, and partition sets, the value returned is blanks.

REMOTESYSTEM (*data-area*) (**programs only**)

Returns the 4-character name of the CICS region in which the module is defined (from the REMOTESYSTEM value in the PROGRAM definition). It applies only to programs, and only to those defined to be remote; for local programs, map sets, and partition sets, the value returned is blanks.

RESCOUNT (*data-area*)

Returns a fullword binary field giving the number of separate uses of this module that are taking place at the time of this inquiry. A value of -1 is returned if the module is either a remote program, or a Java program that runs in a JVM.

RUNTIME (*cvda*)

Returns a CVDA value indicating the runtime environment of the program. CVDA values are as follows:

JVM The program is a Java program that runs in a Java Virtual Machine (JVM).

LENV The program will run with Language Environment runtime support.

NONLE370

The program will run with a language-specific runtime environment.

NOTAPPLIC

RUNTIME does not apply because the module is a map set or a partition set.

UNKNOWN

The program runtime environment is unknown, because the program has not been loaded by CICS, and therefore its source language has not been deduced, which dictates the runtime environment to be used.

XPLINK

The program is a C or C++ program which has been compiled using the XPLINK option.

SHARESTATUS (*cvda*)

Returns a CVDA value indicating where CICS obtains the module the next time a new copy is required. CVDA values are as follows:

NOTAPPLIC

SHARESTATUS does not apply because the module is a remote program or a Java program that runs in a JVM.

PRIVATE

The module is loaded from one of the libraries in the DFHRPL or dynamic LIBRARY concatenation.

SHARED

The LPA copy is to be used, if one is available. If it is not, the module is loaded as if SHARESTATUS were PRIVATE.

STATUS (*cvda*)

Returns a CVDA value indicating whether the module is available for use. CVDA values are as follows:

DISABLED

The module is not available for use.

ENABLED

The module is available for use.

TRANSID (*data-area*) (**programs only**)

Returns the 4-character name of the transaction under which this module, which must be a program, runs remotely; that is, the transaction identifier that the remote region assigns to the task created there to execute it when a task in the local region links to it. This value comes from the TRANSID option value in the PROGRAM definition and applies only to programs defined as remote;

for local programs, map sets, and partition sets, and when no TRANSID is specified for a remote program, the value returned is blanks.

USECOUNT (*data-area*)

Returns a fullword binary field giving the total number of times the module has been used since the start of the current CICS session. The use count is provided for all modules including Java programs, except for remote programs. A value of -1 is returned if the program is remote.

Conditions

END

RESP2 values:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

PGMIDERR

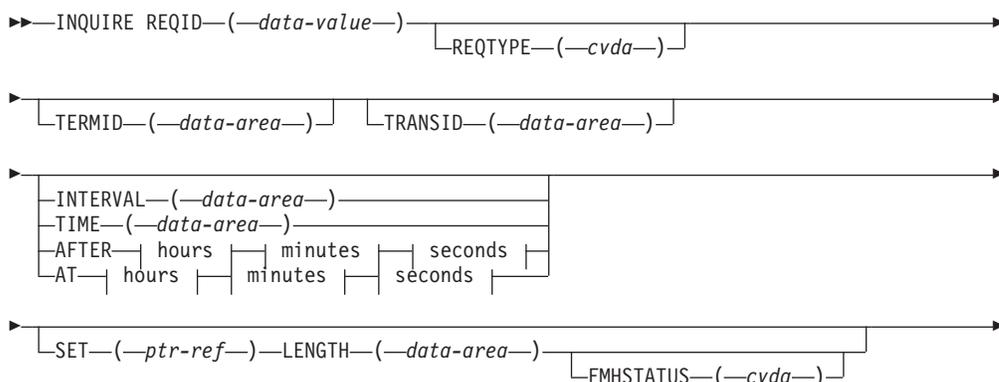
RESP2 values:

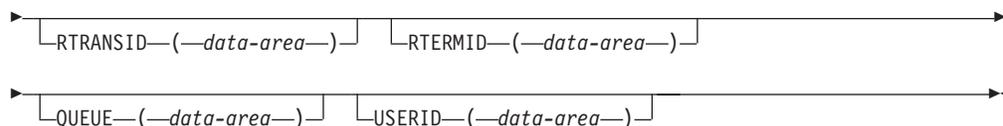
- 1 The program cannot be found. If this error occurs on an INQUIRE PROGRAM NEXT command, an earlier cataloging error has made a PROGRAM, MAPSET, or PARTITIONSET definition unusable, and the definition must be discarded and reinstalled.

INQUIRE REQID

Retrieve information about a queued request.

INQUIRE REQID





hours:



minutes:



seconds:



Conditions: END, ILLOGIC, INVREQ, NOTAUTH, NOTFND

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The INQUIRE REQID command returns information about a queued request. A queued request results from a DELAY, POST, ROUTE, or START command with a nonzero expiry time, and it lasts until that time. For a DELAY command, expiry time is the end of the delay; for a POST, it is the time at which posting is to occur; for a ROUTE, it is the time at which the message is to be delivered; and for a START, it is the time at which CICS is to create the requested task.

After a request expires, you cannot inquire about it with INQUIRE REQID, even if the action requested is not complete. For example, a request to START a transaction may be delayed beyond expiry time, waiting for the terminal it requires.

Requests are identified by the REQID value in the originating command (or assigned by CICS, if omitted in the command). REQID values should be and normally are unique; however, if there is more than one queued request with the same identifier, INQUIRE REQID returns information about the one that will expire soonest.

Expiry time can be expressed either as an interval (the length of time from your INQUIRE to expiry) or as an absolute value (the length of time after the midnight previous to your INQUIRE). If expiry is before midnight of the current day, absolute time is the same as time-of-day, using a 24-hour clock. You can request either form, regardless of how the time was specified in the command that created the request.

There are also two formats for expiry time, whether it is an absolute value or an interval:

- A 4-byte packed decimal composite (0hhmss+), which you obtain by using the TIME or INTERVAL option.
- Separate hours, minutes, and seconds, which you obtain by specifying HOURS, MINUTES, and SECONDS with either AT or AFTER.

Expiry time and request type (the type of command that produced it) are available for any queued request. For START requests additional information is available, including data passed from the starting to the started task.

START commands have four options for passing data. The FROM option is primary, and allows you to pass data of variable length, but three others—QUEUE, RTERMID, and RTRANSID—allow you to pass small items of fixed length. They are intended for convenience in conveying resource names to the started transaction, but are not restricted to that purpose. All four data items are kept in temporary storage, and consequently are subject to explicit deletion by another task. If data that you request in an INQUIRE REQID command has been deleted from temporary storage or cannot be read because of an I/O error, CICS raises the INVREQ condition.

Browsing

You also can browse through all of the queued requests by using the browse options (START, NEXT, and END) on INQUIRE REQID commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

Options

AFTER

requests that CICS return the expiry time (in the HOURS, MINUTES, and SECONDS options) as the **interval** between the current time and the expiry time.

AT

requests that CICS return the expiry time (in the HOURS, MINUTES, and SECONDS options) as an **absolute** value (following the midnight preceding this inquiry).

FMHSTATUS(*cvda*)

returns a CVDA value indicating whether the data passed in the FROM option of the command that created this request contains function management headers. FMHSTATUS applies only to requests resulting from ROUTE commands, or START commands that specify FROM. CVDA values are:

FMH The data contains a function management header.

NOFMH

The data does not contain a function management header.

NOTAPPLIC

The request did not result from a ROUTE or START command, or there was no FROM data.

HOURS(*data-area*)

returns a fullword binary field giving the hours portion of the expiry time (see the AT and AFTER options).

INTERVAL(*data-area*)

returns the expiry time as an interval from the current time. The value is a 4-byte packed decimal number in the format 0hhmss+.

LENGTH(*data-area*)

returns a halfword binary field giving the length of the data passed in the FROM option of the command that created this request. It applies only to requests resulting from ROUTE commands, or START commands that specify FROM; for other requests, the value returned is zero.

MINUTES(*data-area*)

returns a fullword binary field giving the minutes portion of the expiry time (see the AT and AFTER options).

QUEUE(*data-area*)

returns the 8-byte field passed in the QUEUE option of the START command that created this request. It applies only to requests resulting from START commands that specify QUEUE; for other requests, the value returned is blanks.

REQID(*data-value*)

specifies the 8-byte identifier of the request about which you are inquiring. This is the value specified in the REQID option of the command that generated the request (or assigned by CICS if REQID was omitted).

REQTYPE(*cvda*)

returns a CVDA value indicating the type of command that created this request. CVDA values are:

DELAY

A DELAY command created this request.

POST A POST command created this request.

ROUTE

A ROUTE command created this request.

START

A START command created this request.

RTERMID(*data-area*)

returns the 4-byte field passed in the RTERMID option of the START command that created this request. It applies only to requests resulting from START commands that specify RTERMID; for other requests, the value returned is blanks.

RTRANSID(*data-area*)

returns the 4-byte field passed in the RTRANSID option of the START command that created this request. It applies only to requests resulting from START commands that specify RTRANSID; for other requests, the value returned is blanks.

SECONDS(*data-area*)

returns a fullword binary field giving the seconds portion of the expiry time (see the AT and AFTER options).

SET(*ptr-ref*)

returns the address of the data passed in the FROM option of the command which created this request. It applies only to requests resulting from ROUTE commands, or START commands that specify FROM; for other requests, the value returned is the null pointer (X'FF000000').

TERMID(*data-area*)

returns the 4-character terminal identifier that was specified in the TERMID

option of the START command that created the request. It applies only to requests originating from START commands that specify a terminal; for other requests, the value returned is blanks.

TIME(*data-area*)

returns the expiry time as an absolute value measured from the midnight preceding this INQUIRE command. The value is a 4-byte packed decimal number in the format 0hhmmss+.

TRANSID(*data-area*)

returns the 4-character transaction identifier that was specified in the TRANSID option of the command that created the request. It applies only to requests originating from ROUTE or START commands; for other requests, the value returned is blanks.

USERID(*data-area*)

returns the 8-character identifier of the user associated with the task that issued the command that created this request. USERID applies only to requests resulting from ROUTE or START commands.

For a START command:

- if a TERMID is specified on the START command, the value returned is blanks,
- if a USERID is specified on the START command, that user-id is returned,
- if neither of these is specified on the START command, the user-id of the task that issued the START command is returned.

For other requests, the value returned is blanks.

Conditions

END

RESP2 values:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

INVREQ

RESP2 values:

- 3 An I/O error occurred while an attempt was made to read data from temporary storage for the SET, QUEUE, RTERMID, or RTRANSID option.
- 4 Data required for the SET, QUEUE, RTERMID, or RTRANSID option cannot be returned because it has been deleted from temporary storage.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

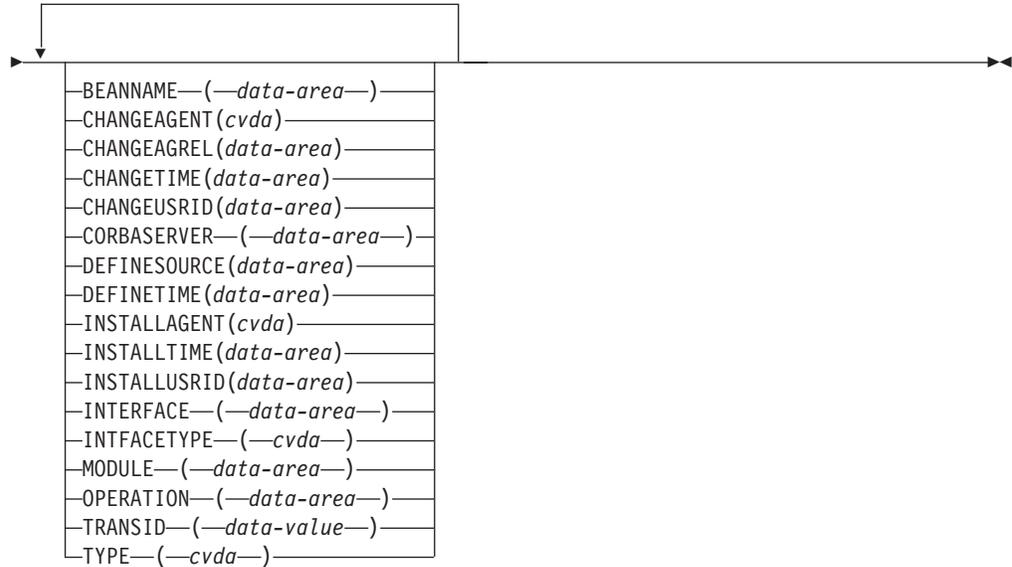
RESP2 values:

INQUIRE REQUESTMODEL

Retrieve information about a REQUESTMODEL.

INQUIRE REQUESTMODEL

►►—INQUIRE REQUESTMODEL—(—*data-value*—)—————►



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

Description

A REQUESTMODEL resource definition maps an inbound request that is formatted using the Internet Inter-ORB PROTOCOL (IIOP) to a CICS transaction that is to be started to process the request. The INQUIRE REQUESTMODEL command returns information about an installed REQUESTMODEL.

Browsing

You can also browse through all the REQUESTMODELs that are installed on your system by using the browse options (START, NEXT, and END) on INQUIRE REQUESTMODEL commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples. You can specify the following options on the INQUIRE request:

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID,

DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

REQUESTMODEL (*data-value*)

Specifies an 8-character request model name.

BEANNAME (*data-area*)

Returns the 240-character field containing the (possibly generic) bean name that matches the name of an enterprise bean in an XML deployment descriptor. This field is blank if the TYPE attribute is CORBA.

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see FORMATTIME in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

CORBASERVER (*data-area*)

Returns a 4-character field containing the (possibly generic) name of the destination CorbaServer for this REQUESTMODEL.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID (*data-area*)

Returns the 8-character user ID that installed the resource.

INTERFACE (*data-area*)

Returns a (possibly generic) name, of up to 255 characters, matching the IDL interface name. This field is blank if the TYPE attribute is EJB.

INTFACETYPE (*cvda*)

Returns the CVDA value of the Java interface type for this REQUESTMODEL. CVDA values are as follows:

HOME

Specifies that this interface is the home interface for the bean.

REMOTE

Specifies that this interface is the component interface for the bean.

BOTH Matches both the home and component interfaces for the bean.

NOTAPPLIC

Returned if the TYPE attribute is CORBA.

MODULE (*data-area*)

Returns a (possibly generic) name, of up to 255 characters, matching the IDL module name. This field is blank if the TYPE attribute is EJB.

OPERATION (*data-area*)

Returns a 255-character field containing a (possibly generic) name matching the IDL operation or bean method name.

TRANSID (*name*)

Returns the 4-character name of the CICS transaction to be run when a request matching the specification of the REQUESTMODEL is received.

TYPE (*data-area*)

Returns a CVDA value indicating the type of the REQUESTMODEL. CVDA values are as follows:

EJB Matches enterprise bean requests as specified by the EJB parameters.

CORBA

Matches CORBA requests as specified by the CORBA parameters.

GENERIC

Matches both enterprise bean and CORBA requests.

Conditions**END**

RESP2 values:

- 2 All authorized resource definitions have been retrieved. All data areas specified on this command are left unchanged.

ILLOGIC

RESP2 values:

- 1 A START has been given when a browse is already in progress, or a NEXT or END has been given without a preceding START.

NOTAUTH

RESP2 values:

- 100 The user is not authorized for this command.

NOTFND

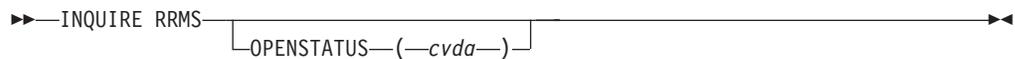
RESP2 values:

- 1 The specified request model was not found.

INQUIRE RRMS

Retrieves the status of transactional EXCI.

INQUIRE RRMS



Conditions: NOTAUTH

Description

The **INQUIRE RRMS** command indicates whether inbound transactional EXCI work is currently being accepted.

Options

OPENSTATUS(*cvda*)

returns a CVDA value indicating whether CICS accepts inbound transactional EXCI work or not. CVDA values are:

OPEN indicates that CICS does accept inbound transactional EXCI work.

CLOSED

indicates that CICS does not accept inbound transactional EXCI work.

NOTAPPLIC

indicates that CICS has been initialized without RRMS.

Conditions

NOTAUTH

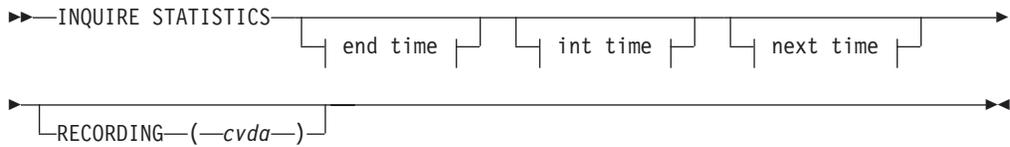
RESP2 values:

- 100 The user is not authorized for this command.

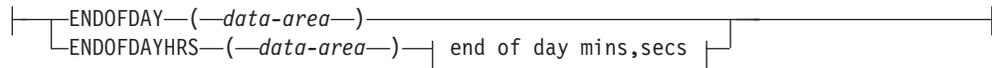
INQUIRE STATISTICS

Retrieve statistics information.

INQUIRE STATISTICS



end time:



end of day mins,secs:



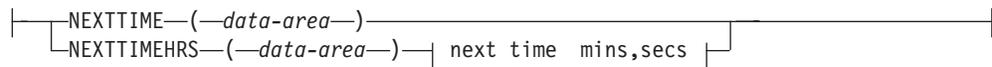
int time:



interval mins,secs:



next time:



next time mins,secs:



Conditions: NOTAUTH

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The INQUIRE STATISTICS command returns information about the recording of CICS resource and system statistics. CICS records system statistics periodically if the RECORDING switch is on, at a frequency governed by the INTERVAL option. These statistics are called **interval statistics**. At end-of-day time (the ENDOFDAY option), CICS records **end-of-day statistics**—which are the statistics for the interval since the last resetting—whether or not the switch is on, ensuring that statistics are written at least once a day. Recording occurs on a system management facility (SMF) data set, and the counts are reset after recording.

There are two formats for each of the time values that you can retrieve with this command (the end-of-day time, the recording interval, and the next time that recording will occur):

- A 4-byte packed decimal composite (0hhmmss+), which you obtain by using the ENDOFDAY, INTERVAL, and NEXTTIME options.
- Separate hours, minutes, and seconds, which you obtain by specifying the ENDOFDAYHRS, ENDOFDAYMINS, and ENDOFDAYSECS options (instead of ENDOFDAY), INTERVALHRS, INTERVALMINS, and INTERVALSECS (instead of INTERVAL) and NEXTTIMEHRS, NEXTTIMEMINS, and NEXTTIMESECS (instead of NEXTTIME).

The *CICS Performance Guide* contains more detail about CICS statistics, and the description of the “SET STATISTICS” on page 708 command describes the relationship between the interval and end-of-day times.

Options

ENDOFDAY (*data-area*)

returns the end-of-day time, as a 4-byte packed decimal field in the format 0hhmmss+. End-of-day time is expressed in local time.

ENDOFDAYHRS (*data-area*)

returns the hours component of the end-of-day time, in fullword binary form.

ENDOFDAYMINS (*data-area*)

returns the minutes component of the end-of-day time, in fullword binary form.

ENDOFDAYSECS (*data-area*)

returns the seconds component of the end-of-day time, in fullword binary form.

INTERVAL (*data-area*)

returns a 4-byte packed decimal field giving the recording interval for system statistics.

INTERVALHRS (*data-area*)

returns the hours component of the recording interval, in fullword binary form.

INTERVALMINS (*data-area*)

returns the minutes component of the recording interval, in fullword binary form.

INTERVALSECS (*data-area*)

returns the seconds component of the recording interval, in fullword binary form.

NEXTTIME (*data-area*)

returns a 4-byte packed decimal field giving the time at which statistics are recorded next (assuming that the RECORDING switch is not changed from its current value). This is the end-of-day time if RECORDING is currently off, and the earlier of end-of-day and the end of the current interval otherwise.

NEXTTIMEHRS (*data-area*)

returns the hours component of the next recording time, in fullword binary form.

NEXTTIMEMINS (*data-area*)

returns the minutes component of the next recording time, in fullword binary format.

NEXTTIMESECS (*data-area*)

returns the seconds component of the next recording time, in fullword binary format.

RECORDING (*cvda*)

controls the recording of interval statistics, End-of-day statistics, requested statistics and unsolicited statistics are always recorded, irrespective of the setting of the RECORDING option. (Unsolicited statistics are resource statistics, recorded when the resource is discarded. Requested statistics are those called for by a "PERFORM STATISTICS RECORD" on page 598 command, or by a CEMT PERFORM STATISTICS transaction.)

CVDA values are:

OFF switches off the recording of interval statistics.

ON switches on the recording of interval statistics.

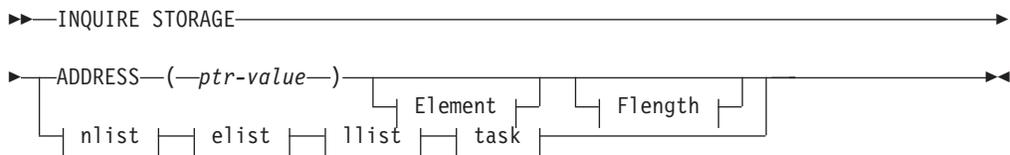
Conditions**NOTAUTH**

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

INQUIRE STORAGE

Retrieve information about task storage.

INQUIRE STORAGE**Element:****Flength:****nlist:****elist:**

l|list:

```
|-----|
|_LENGTHLIST--(ptr-ref)_|
```

task:

```
|-----|
|_TASK--(data-value)_|
```

Conditions: INVREQ, NOTAUTH, TASKIDERR

Description

The INQUIRE STORAGE command has two functions. You can use it to get a list of the task storage areas associated with a particular task (using the NUMELEMENTS option), or you can use it to find the length and starting address of a particular area of storage (using the ADDRESS option). INQUIRE STORAGE applies only to storage allocated to user tasks, which are tasks executing user-defined transactions or the CICS-supplied transactions normally invoked by an operator.

Options**ADDRESS** (*ptr-value*)

specifies that you are inquiring about a single area of storage and identifies the area. The address you specify can be anywhere within the area about which you are inquiring; it does not have to be the start of it. CICS returns the length of the area (in FLENGTH) and its starting address (in ELEMENT) if it is a valid element of user task storage.

DSANAME (*data-value*)

specifies the name of the DSA for which storage elements are to be returned.

Possible values are CDSA, UDSA, ECDSA, and EUDSA. If you omit this option, storage elements are returned for all four DSAs.

ELEMENT (*ptr-ref*)

returns the starting address of the storage area containing the address provided in the ADDRESS option, if the area is user task storage. This is the first byte of the area available for task data, not the preceding storage management control information, if any. If the area is not user task storage, the address returned is nulls.

ELEMENTLIST (*ptr-ref*)

returns the address of a list of the addresses of all areas of task storage for the task specified in the TASK option. Each address points to the first byte available for data storage, not to preceding storage management control information, if any. The number of addresses in this list is the NUMELEMENTS option value. (Addresses are 4 bytes long, and therefore the length of the list in bytes is 4 times NUMELEMENTS.)

CICS obtains the storage for this list and frees it when the inquiring task ends, or issues another INQUIRE STORAGE command with ELEMENTLIST or LENGTHLIST, or issues an INQUIRE TASK LIST; the task cannot free the storage itself.

FLENGTH(*data-area*)

returns a fullword binary field giving the length of the storage area containing the address provided in the ADDRESS option. This is the length of the part available for task data; it does not include storage management control information at the beginning or end of the area, if any. If the area is not user task storage, the length returned is -1.

LENGTHLIST(*ptr-ref*)

returns the address of a list of fullword binary lengths. Each entry in this list is the length of the storage area to which the corresponding entry in the ELEMENTLIST list points. These lengths are the amounts available for data storage and do not include storage management control information, if any.

CICS obtains the storage for this list and frees it when the inquiring task ends, or issues another INQUIRE STORAGE command with ELEMENTLIST or LENGTHLIST, or issues an INQUIRE TASK LIST; the task cannot free the storage itself.

NUMELEMENTS(*data-area*)

indicates that you are requesting a list of the task storage areas for the task indicated in the TASK option. CICS returns the number of areas, in fullword binary form, in the data area you provide. If you request an ELEMENTLIST or LENGTHLIST, this value is the number of entries in the list.

TASK(*data-value*)

specifies, as a 4-byte packed decimal value, the task number for which you are requesting a storage list. If you omit this option but include NUMELEMENTS, CICS assumes the inquiry is for the task issuing the INQUIRE STORAGE command.

Conditions**INVREQ**

RESP2 values:

- 1 Invalid DSANAME specified.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

TASKIDERR

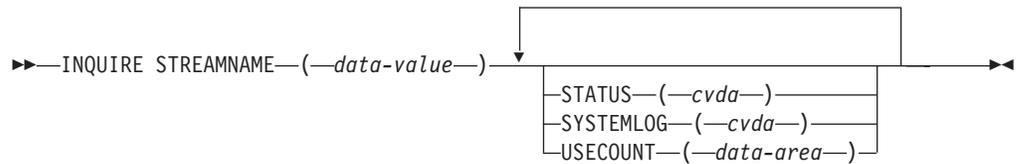
RESP2 values:

- 1 The task number does not exist.
- 2 The task number designates a system task, not a user task.

INQUIRE STREAMNAME

Retrieve information about a currently connected MVS log stream.

INQUIRE STREAMNAME



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

Description

The INQUIRE STREAMNAME command allows you to look at information about a currently connected MVS log stream.

Browsing

You can also browse through log stream names by using the browse options (START, NEXT, and END) on INQUIRE STREAMNAME commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

Options

STATUS(*cvda*)

returns a CVDA value indicating the status of the log stream. CVDA values are:

FAILED

The message logger has detected a problem with the specified log stream.

OK No errors have been detected.

STREAMNAME(*data-value*)

specifies an MVS system logger log stream name.

CICS returns a NOTFND condition if the log stream name does not exist, or if there are no longer any users of the log stream in this CICS region (see the USECOUNT option).

SYSTEMLOG(*cvda*)

returns a CVDA value indicating whether the log stream is the system log. CVDA values are:

NOSYSLOG

The log stream is not the system log.

SYSLOG

The log stream is the system log.

USECOUNT(*data-area*)

returns the number of CICS journal names and forward recovery logs within this CICS system that are currently using the log stream.

The use count is always at least 1, because CICS does not maintain any information about a log stream that no longer has any users, in which case an INQUIRE STREAMNAME command returns a NOTFND condition.

See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

Options

DSANAME (*data-area*)

Returns an 8-character field giving the name of the dynamic storage area (DSA) in which the specified subpool resides. The value can be one of the following, padded with trailing blanks (X'40'):

CDSA
ECDSA
ERDSA
ESDSA
ETDSA
GCDSA
RDSA
SDSA

SUBPOOL (*data-area*)

Specifies the 8-character name of a storage subpool. For a full list of all storage subpools that can exist in a CICS region, see CICS subpools in the CICS Performance Guide.

For browse operations, specify SUBPOOL on the START browse request only, not on the NEXT or END requests.

Conditions

END

RESP2 values:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

- 1 The subpool name specified on the command does not exist.

INQUIRE SYSDUMPCODE

Retrieve information about a system dump table entry.

INQUIRE SYSDUMPCODE

NODAE

The dump is not eligible for DAE suppression—if CICS determines that a dump should be written, MVS does not suppress it. (However, be aware of the SUPPRESS and SUPPRESSALL options in the ADYSETxx parmlib member. These are controlled by the VRADAE and VRANODAE keys in the SDWA. They may lead to dump suppression even though NODAE is set here. For information about DAE, SUPPRESS, and SUPPRESSALL, see *MVS Diagnosis: Tools and Service Aids*.

DUMPSCOPE(*cvda*)

returns a CVDA value indicating whether a request for a dump with this dump code should cause an SDUMP (system dump) request to be sent to related MVS images.

A related MVS image is one that contains a CICS region doing work on behalf of your CICS region. Specifically, it is a region that has one or more tasks doing work under the same APPC token as a task in your region.

The sending of SDUMP requests occurs only when the table entry for this code specifies a dump (that is, the SYSDUMPING value is SYSDUMP), and only in a sysplex environment executing under MVS/ESA Version 5 or later and the MVS workload manager.

CVDA values are:

LOCAL

SDUMP requests are not to be sent.

RELATED

SDUMP requests are to be sent.

Note: A setting of DUMPSCOPE(RELATED) results in a single dump being taken for each affected MVS image. This dump contains the output from all the affected CICS regions in the image. For more information, see Automatic dump data capture from related CICS regions, in the *CICS Problem Determination Guide*.

MAXIMUM(*data-area*)

returns a fullword binary field giving the maximum number of dumps with this code that CICS will take. A value of 999 means the default, 'no limit'.

SHUTOPTION(*cvda*)

returns a CVDA value indicating whether the CICS system is to be shut down after a request for a dump with this dump code. CVDA values are:

NOSHUTDOWN

The CICS system is not to be shut down.

SHUTDOWN

The CICS system is to be shut down.

SYSDUMPCODE(*data-value*)

specifies the 8-character system dump code about which you are inquiring. A valid code contains no leading or imbedded blanks.

SYSDUMPING(*cvda*)

returns a CVDA value indicating whether a dump request with this code should produce a dump or not. Even when a dump is specified, CICS will take one only when the CURRENT value is no greater than the MAXIMUM, and when system dumps are not suppressed globally (see the DUMPING option of

the "INQUIRE SYSTEM" command. MVS may also be allowed to suppress the dump if appropriate (the DAE option). CVDA values are:

NOSYSDUMP

A dump is not to be taken.

SYSDUMP

A dump is to be taken.

Note: Dumps from the kernel domain of CICS are not subject to suppression and are taken regardless of SYSDUMPCODE value.

Conditions

END

RESP2 values:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

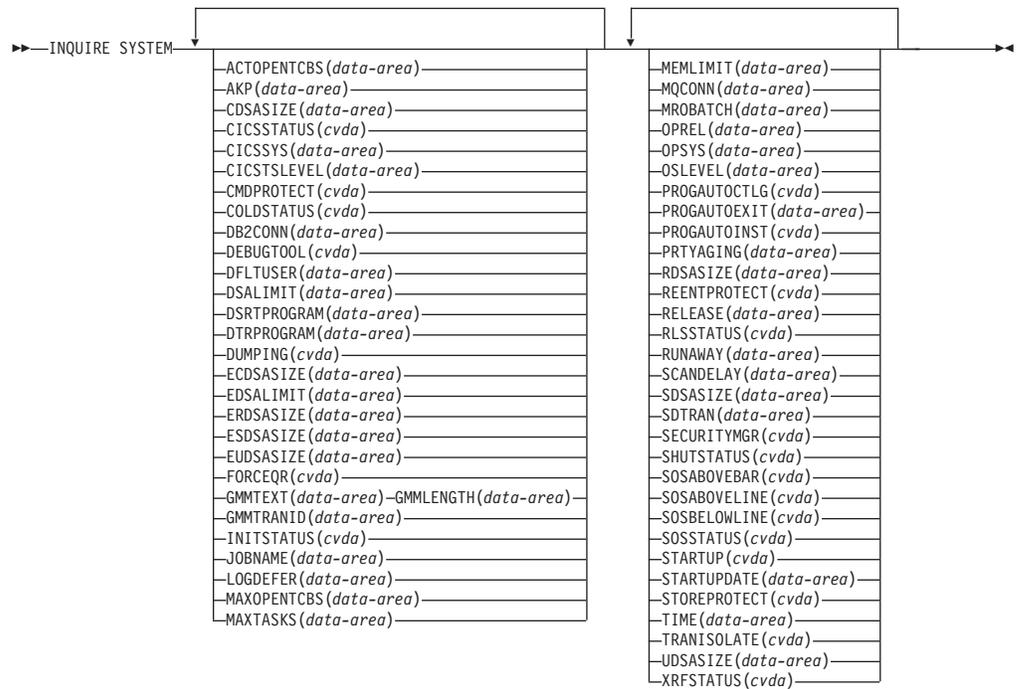
RESP2 values:

- 1 The named dump code cannot be found.

INQUIRE SYSTEM

Retrieve CICS system information.

INQUIRE SYSTEM



Condition: NOTAUTH

For more information about the use of CVDAs, see CICS-value data areas (CVDA) in CICS System Programming Reference.

Description

The **INQUIRE SYSTEM** command returns information about the CICS system in which the task that issues the command is running.

Many options in this command correspond to system initialization parameters and take their initial values from the parameters. Some options can be changed by a subsequent **SET SYSTEM** command, or its **CEMT** equivalent. Other options return information about the CICS or MVS release levels, and others return information determined solely by the current state of the system. Table 4 indicates the origin of the option values and, for options that are specified in the system initialization parameter, the name of the parameter.

Table 4. INQUIRE SYSTEM options

Option	Origin
ACTOPENTCBS	System state
AKP	AKPFREQ system initialization parameter
CDSASIZE	System state
CICSSTATUS	System state
CICSSYS	System state
CICSTSLEVEL	CICS control block
CMDPROTECT	CMDPROT system initialization parameter
COLDSTATUS	System state
DB2CONN	Installed DB2CONN resource definition
DEBUGTOOL	DEBUGTOOL system initialization parameter
DFLTUSER	DFLTUSER system initialization parameter
DSALIMIT	DSALIM system initialization parameter

Table 4. INQUIRE SYSTEM options (continued)

Option	Origin
DSRTPROGRAM	DSRTPGM system initialization parameter
DTRPROGRAM	DTRPGM system initialization parameter
DUMPING	DUMP system initialization parameter
ECDSASIZE	System state
EDSALIMIT	EDSALIM system initialization parameter
ERDSASIZE	System state
ESDSASIZE	System state
EUDSASIZE	System state
FORCEQR	FORCEQR system initialization parameter
GMMTEXT, GMMLLENGTH	GMTEXT system initialization parameter
GMMTRANID	GMTRAN system initialization parameter
INITSTATUS	System state
JOBNAME	JCL or cataloged procedure
LOGDEFER	LGDFINT system initialization parameter
MAXOPENTCBS	MAXOPENTCBS system initialization parameter
MAXTASKS	MXT system initialization parameter
MEMLIMIT	System state
MQCONN	Installed MQCONN resource definition
MROBATCH	MROBTCH system initialization parameter
OPREL	Operating system (MVS)
OPSYS	Operating system (MVS)
OSLEVEL	Operating system (z/OS)
PROGAUTOCTLG	PGAICTLG system initialization parameter
PROGAUTOEXIT	PGAIEEXIT system initialization parameter
PROGAUTOINST	PGAIPGM system initialization parameter
PRTYAGING	PRTYAGE system initialization parameter
RDSASIZE	System state
REENTPROTECT	RENTPGM system initialization parameter
RELEASE	CICS system code
RLSSTATUS	RLS system initialization parameter
RUNAWAY	ICVR system initialization parameter
SCANDELAY	ICVTSD system initialization parameter
SDSASIZE	System state
SDTRAN	SDTRAN system initialization parameter
SECURITYMGR	SEC system initialization parameter
SHUTSTATUS	System state
SOSABOVEBAR	System state
SOSABOVELINE	System state
SOSBELOWLINE	System state
SOSSTATUS	System state
STARTUP	System state
STARTUPDATE	System state
STOREPROTECT	STGPROT system initialization parameter, and hardware
TIME	ICV system initialization parameter
TRANISOLATE	TRANISO system initialization parameter, and hardware
UDSASIZE	System state
XRFSTATUS	XRF system initialization parameter, and system state

Note: The CSCS, ECSCS, ERSCS, EUSCS, and USCS options, each of which returned the size of the storage cushion for a particular dynamic storage area, are obsolete in CICS Transaction Server for z/OS. The translator accepts them and gives a warning. At run time, the data areas provided are left unchanged.

Options

ACTOPENTCBS (*data-area*)

Returns a fullword binary field giving the number of open TCBs currently allocated to user tasks. The open TCBs are allocated from the pool of open TCBs that CICS attaches, up to the maximum set by the MAXOPENTCBS system initialization parameter. The ACTOPENTCBS value can be equal to, or less than, the MAXOPENTCBS value. If the value is equal to MAXOPENTCBS, tasks that require an open TCB are made to wait.

AKP (*data-area*)

Returns a fullword binary field giving the activity keypoint trigger value, which is the number of write requests to the CICS system log stream output buffer between the taking of keypoints.

A value of minus one (not applicable) means that keypoints are not being taken.

CDSASIZE (*data-area*)

Returns the current size in bytes of the CICS dynamic storage area (CDSA), in fullword binary form. It includes both storage in use and storage available for use. This size is calculated and managed by CICS automatically, within the DSALIMIT value, that is, the overall limit for dynamic storage areas that reside below 16 MB (below the line).

CICSSTATUS (*cvda*)

Returns a CVDA value that identifies the current execution status of CICS:

ACTIVE

CICS is fully active.

FINALQUIESCE

CICS is in the final quiesce stage of shutdown. Programs in the second stage of the program list table for shutdown (PLTSD) are run during this stage.

FIRSTQUIESCE

CICS is in the first quiesce stage of shutdown. Programs in the first stage of the PLTSD are run during this stage.

STARTUP

CICS is starting up but is not yet fully active. Programs in the program list table for program initiation (PLTPI) are run during startup. See the INITSTATUS option for further information.

CICSSYS (*data-area*)

Returns a 1-character value that identifies the operating system for which the running CICS system has been built. A value of "X" represents MVS.

CICSTSLEVEL (*data-area*)

Returns a 6-character value that identifies the version, release, and modification level of the CICS Transaction Server for z/OS product under which the CICS region is running. The value is of the form *vrrmm*, and CICS Transaction Server for z/OS, Version 4 Release 2 returns 040200.

CMDPROTECT (*cvda*)

Returns a CVDA value that indicates whether command protection is active.

When a task issues a command and command protection is active, CICS verifies that the task has write access to the first byte of every area into which CICS is to return information. If any area fails the test, an AEYD abend occurs.

The CVDA values are as follows:

CMDPROT

Command protection is active.

NOCMDPROT

Command protection is not active.

COLDSTATUS (*cvda*)

Returns a CVDA value that indicates whether CICS performed a cold or an initial start. (For information about the types of CICS startup, see the *CICS Recovery and Restart Guide*.)

The CVDA values are as follows:

COLD CICS performed a cold start. Log information about local resources was erased, but information about the outcome of local units of work, needed to allow remote systems or RMI-connected resource managers to resynchronize their resources, was preserved.

INITIAL

CICS performed an initial start. All log information about both local and remote resources was erased.

NOTAPPLIC

CICS performed neither a cold nor an initial start.

DB2CONN (*data-area*)

Returns the 1-8 character name of the DB2CONN resource definition that is currently installed for the CICS region, or blanks if no DB2CONN definition is currently installed. Only one DB2CONN definition can be installed at a time. The DB2CONN resource definition specifies the attributes of the connection between CICS and DB2.

DEBUGTOOL (*cvda*)

Returns a CVDA value that indicates whether you can use debugging profiles to select the programs that will run under the control of a debugging tool. The following debugging tools use debugging profiles:

- Debug Tool, for compiled language application programs (programs written in COBOL, PL/I, C, C++, and Assembler)
- Remote debugging tools (for compiled language application programs and Java programs)

Other debugging mechanisms, for example the CICS Execution Diagnostic Facility (CEDF), do not use debugging profiles. The CVDA values are as follows:

DEBUG

You can use CICS debugging profiles to select the programs that will run under the control of a debugging tool.

NODEBUG

You cannot use CICS debugging profiles to select the programs that will run under the control of a debugger tool.

For more information about using debugging profiles, see the *CICS Application Programming Guide*.

DFLTUSER(*data-area*)

Returns the 8-character identifier of the default user for this CICS region.

DSALIMIT(*data-area*)

Returns a fullword binary field giving the maximum amount of storage, in bytes, within which CICS can dynamically allocate storage for the four individual dynamic storage areas that reside below 16 MB (below the line). See the CDSASIZE, RDSASIZE, SDSASIZE, and UDSASIZE options of this command.

DSRTPROGRAM(*data-area*)

Returns the 8-character name of the distributed routing program.

DTRPROGRAM(*data-area*)

Returns the 8-character name of the dynamic routing program.

DUMPING(*cvda*)

Returns a CVDA value that indicates whether the taking of CICS system dumps is suppressed:

NOSYSDUMP

System dumps are suppressed.

SYSDUMP

System dumps are not suppressed.

These values are set by the system initialization parameter DUMP=YES or NO.

ECDSASIZE(*data-area*)

Returns the current size in bytes of the extended CICS dynamic storage area (ECDSA), in fullword binary form. It includes both storage in use and storage available for use. This size is calculated and managed by CICS automatically, within the EDSALIMIT value, that is, the overall limit for dynamic storage areas that reside above 16 MB but below 2 GB (above the line).

EDSALIMIT(*data-area*)

Returns a fullword binary field giving the maximum amount of storage, in bytes, within which CICS can dynamically allocate storage for the individual dynamic storage areas that reside above 16 MB but below 2 GB (above the line). See the ECDSASIZE, ERDSASIZE, ESDSASIZE, and EUDSASIZE options of this command.

ERDSASIZE(*data-area*)

Returns the current size in bytes of the extended read-only dynamic storage area (ERDSA), in fullword binary form. It includes both storage in use and storage available for use. This size is calculated and managed by CICS automatically, within the EDSALIMIT value, that is, the overall limit for dynamic storage areas that reside above 16 MB but below 2 GB (above the line).

ESDSASIZE(*data-area*)

Returns the current size in bytes of the extended shared dynamic storage area (ESDSA), in fullword binary form. It includes both storage in use and storage available for use. This size is calculated and managed by CICS automatically, within the EDSALIMIT value, that is, the overall limit for dynamic storage areas that reside above 16 MB but below 2 GB (above the line).

EUDSASIZE(*data-area*)

Returns the current size in bytes of the extended user dynamic storage area (EUDSA), in fullword binary form. It includes both storage in use and storage available for use. This size is calculated and managed by CICS automatically,

within the EDSALIMIT value, that is, the overall limit for dynamic storage areas that reside above 16 MB but below 2 GB (above the line).

FORCEQR(*cvda*)

Returns a CVDA value that indicates whether CICS forces CICSAPI user application programs that are defined as threadsafe to run on the quasi-reentrant (QR) TCB:

FORCE

CICS forces all user application programs specified with the CONCURRENCY(THREADSAFE) attribute to run under the QR TCB, as if they were specified with CONCURRENCY(QUASIRENT). Force does not apply to certain programs, for example OPENAPI programs, or C or C++ programs compiled with XPLINK. For details, see FORCEQR system initialization parameter in the System Definition Guide.

NOFORCE

CICS honors the CONCURRENCY(THREADSAFE) attribute on CICSAPI user application programs, and allows user programs to run on an open TCB to avoid unnecessary TCB switching.

GMMLENGTH(*data-area*)

Returns a halfword binary field that shows the length in bytes of the “good morning” message text.

GMMTEXT(*data-area*)

Returns the “good morning” message text in the data-area you provide, which must be long enough to accommodate it. The maximum length of any “good morning” message is 246 bytes. The actual length is returned in the GMMLENGTH option value.

GMMTRANID(*data-area*)

Returns the 4-character name of the transaction that generates the “good morning” message.

INITSTATUS(*cvda*)

Returns a fullword binary field that shows the initialization status of the CICS system. CVDA values are as follows:

FIRSTINIT

First stage of CICS initialization.

INITCOMPLETE

CICS initialization is complete.

SECONDINIT

Second stage of initialization.

THIRDINIT

Third stage of initialization.

JOBNAME(*data-area*)

Returns the 8-character MVS jobname under which CICS is running.

LOGDEFER(*data-area*)

Returns the halfword binary value that shows the log deferral interval, which is the period of time used by CICS Log Manager when determining how long to delay a forced journal write request before invoking the MVS system logger. See LGDFINT for information about the LOGDEFER parameter and associated system initialization parameter LGDFINT.

MAXOPENTCBS (*data-area*)

Returns a fullword binary field that shows the maximum number of open TCBs that CICS can attach and maintain in its pool of open TCBs. For information about the number allocated, see the ACTOPENTCBS option.

The difference between MAXOPENTCBS and ACTOPENTCBS represents the number of open TCBs that are free.

MAXTASKS (*data-area*)

Returns a fullword binary field that shows the maximum number of tasks that can be eligible for dispatch at any one time in this CICS system. Both active and suspended tasks count toward this limit, but tasks that have not reached the point of initial dispatch do not. System tasks such as terminal and journal control tasks are not counted.

MEMLIMIT (*data-area*)

Returns a doubleword binary field that shows the maximum amount, in bytes, of storage above the bar for use by the CICS region. A value of -1 indicates that no limit has been imposed on the amount of storage that the region can attempt to use (also known as NOLIMIT). The MEMLIMIT value can be set as a PARMLIB member, by JCL, or through the IEFUSI global user exit.

MQCONN (*data-area*)

Returns the 1- to 8-character name of the MQCONN resource definition that is currently installed for the CICS region, or blanks if no MQCONN definition is currently installed. Only one MQCONN definition can be installed at a time. The MQCONN resource definition specifies the attributes of the connection between CICS and WebSphere MQ.

MROBATCH (*data-area*)

Returns a fullword binary field that shows the number of events that must occur, from a list of MRO and DASD I/O events on which CICS is waiting, before CICS is posted explicitly to process them.

OPREL (*data-area*) **(supported for compatibility only)**

Returns a halfword binary field that shows the last 2 digits of the level number of the operating system under which the CICS region is running. For example, z/OS Release 9 is represented by 09.

Note: This field is supported for compatibility purposes only. The information is derived from the last two numbers held in the MVS CVTPRODN field. See the OSLEVEL field for the full version and release number of z/OS.

OPSYS (*data-area*)

Returns a 1-character value that identifies the operating system under which CICS is running. A value of "X" represents MVS.

OSLEVEL (*data-area*)

Returns a 6-byte field that shows the version, release, and modification level of the z/OS product on which CICS is running. For example, z/OS Version 1 Release 9 Modification 0 returns the value 010900.

PROGAUTOCTLG (*cvda*)

Returns a CVDA value that indicates whether and when autoinstalled program definitions are cataloged. Cataloged definitions are restored on a warm or emergency restart. Those not cataloged are discarded at shutdown, and must be installed again if they are used in a subsequent execution of CICS.

Decisions to catalog are made both at initial installation and whenever an autoinstalled definition is modified, and are based on the PROGAUTOCTLG value at the time. CVDA values are as follows:

CTLGALL

Definitions are cataloged both when installed and when modified.

CTLGMODIFY

Definitions are cataloged only when modified.

CTLGNONE

Definitions are not cataloged.

PROGAUTOEXIT(*data-area*)

Returns the 8-character name of the user-provided program that is called by the CICS program autoinstall code to provide a model definition.

PROGAUTOINST(*cvda*)

Returns a CVDA value that indicates whether autoinstall for programs is active or inactive. When a task requests a program, map set, or partition set that is not defined, CICS attempts to create a definition for it automatically if autoinstall for programs is active. If not, CICS raises the PGMIDERR exceptional condition. CVDA values are as follows:

AUTOACTIVE

Autoinstall for programs is active.

AUTOINACTIVE

Autoinstall for programs is not active.

PRTYAGING(*data-area*)

Returns a fullword binary field giving the rate at which CICS increases the priority of a task waiting for dispatch. CICS increases the task priority by 1 after each PRTYAGING milliseconds of wait time without a dispatch.

RDSASIZE(*data-area*)

Returns the current size in bytes of the read-only dynamic storage area (RDSA), in fullword binary form. It includes both storage in use and storage available for use. This size is calculated and managed by CICS automatically, within the DSALIMIT value, that is, the overall limit for dynamic storage areas that reside below 16 MB (below the line).

REENTPROTECT(*cvda*)

Returns a CVDA value that indicates whether storage for reentrant programs (the RDSA and ERDSA) is in key 0 or CICS key. MVS key 0 storage is write protected from programs running in CICS key or user key; programs in CICS key storage are protected only from those running in user key when CICS key and user key are different (that is, when storage protection is active). CVDA values are as follows:

REENTPROT

Read-only DSAs are in key 0 storage.

NOREENTPROT

Read-only DSAs are in CICS-key storage.

RELEASE(*data-area*) **(supported for compatibility only)**

Returns a 4-character string containing the level number of the CICS code. In this release, the value is 0670.

This option is supported only for compatibility with earlier releases. As an exclusive element of CICS Transaction Server for z/OS, CICS does not have a product version and release number of its own. You are recommended to use CICSTSLEVEL to determine the version and release number of CICS Transaction Server.

RLSSTATUS (*cvda*)

Returns a CVDA value that indicates whether VSAM RLS is active - that is, the CICS region is registered (with a currently-open control ACB) with an SMSVSAM address space:

NOTAPPLIC

This CICS region does not support VSAM RLS because of one of the following reasons:

- CICS initialized with RLS=NO as a system initialization parameter
- CICS has forced RLS=NO because the level of VSAM in the MVS in which CICS is running does not support VSAM RLS.

RLSACTIVE

CICS has registered with an SMSVSAM server and VSAM RLS is currently active.

RLSINACTIVE

CICS has registered with an SMSVSAM server, but VSAM RLS is currently inactive due to an SMSVSAM server failure. All RLS requests fail until CICS performs dynamic VSAM RLS restart, which occurs automatically when the SMSVSAM server has restarted.

RUNAWAY (*data-area*)

Returns a fullword binary field that shows the default value for runaway task time. This value is used for any task executing a transaction whose profile does not specify runaway task time (see the RUNAWAY option of the INQUIRE TRANSACTION command).

SCANDELAY (*data-area*)

Returns a fullword binary field giving the maximum number of milliseconds between a user task making a terminal I/O request and CICS dispatching the terminal control task to process it. This value is sometimes called the "terminal scan delay", and is set by the ICVTSD option in the system initialization table.

SDSASIZE (*data-area*)

Returns the current size in bytes of the shared dynamic storage area (SDSA), in fullword binary form. It includes both storage in use and storage available for use. This size is calculated and managed by CICS automatically, within the DSALIMIT value, that is, the overall limit for dynamic storage areas that reside below 16 MB (below the line).

SDTRAN (*data-area*)

Returns the 4-character name of the transaction to run at the beginning of normal or immediate shutdown. This can be the name of a user-supplied transaction, or the CICS-supplied default transaction, CESD.

SECURITYMGR (*cvda*)

Returns a CVDA value that identifies whether an external security manager (such as RACF) is active in the system:

EXTSECURITY

An external security manager is active.

NOSECURITY

No security is being used.

SHUTSTATUS (*cvda*)

Returns a CVDA value that indicates the shutdown status of CICS (see the CICSSTATUS option):

CANCELLED

CICS is canceled.

CONTROLSHUT

CICS is performing a controlled shutdown (that is, a normal shutdown with a warm keypoint).

NOTAPPLIC

CICS is not shutting down.

SHUTDOWN

CICS is performing an immediate shutdown.

SOSABOVEBAR(*cvda*)

Returns a CVDA value that indicates whether CICS is short on storage in the dynamic storage areas above the bar:

NOTSOS

CICS is not short on storage in any of the dynamic storage areas above the bar.

SOS CICS is short on storage in at least one dynamic storage area above the bar.

SOSABOVELINE(*cvda*)

Returns a CVDA value that indicates whether CICS is short on storage in the dynamic storage areas above 16 MB but below 2 GB (above the line):

NOTSOS

CICS is not short on storage in any of the dynamic storage areas above 16 MB but below 2 GB.

SOS CICS is short on storage in at least one dynamic storage area above 16 MB but below 2 GB.

SOSBELOWLINE(*cvda*)

Returns a CVDA value that indicates whether CICS is short on storage in the dynamic storage areas below 16 MB (below the line):

NOTSOS

CICS is not short on storage in any of the dynamic storage areas below 16 MB.

SOS CICS is short on storage in at least one dynamic storage area below 16 MB.

SOSSTATUS(*cvda*)

Returns a CVDA value that indicates whether CICS is short on storage in any of the dynamic storage areas below 2 GB (below the bar):

NOTSOS

CICS is not short on storage in any of the dynamic storage areas below 2 GB.

SOS CICS is short on storage in at least one dynamic storage area below 16 MB, and at least one dynamic storage area above 16 MB but below 2 GB.

SOSABOVE

CICS is short on storage in at least one dynamic storage area above 16 MB but below 2 GB, but is not short on storage in any of the dynamic storage areas below 16 MB.

SOSBELOW

CICS is short on storage in at least one dynamic storage area below 16 MB, but is not short on storage in any of the dynamic storage areas above 16 MB but below 2 GB.

STARTUP(*cvda*)

Returns a CVDA value that indicates how the current execution of CICS started:

COLDSTART

CICS performed an initial or a cold start.

Note: The STARTUP option does not distinguish between an initial and a cold start. See the COLDSTATUS option.

EMERGENCY

CICS performed an emergency restart because the previous run did not shut down normally.

WARMSTART

CICS performed a warm restart following the normal shutdown of the previous run.

STARTUPDATE(*data-area*)

Returns a 4-byte packed-decimal field containing the date on which the current execution of CICS started. The date is in the form *0cyyddd+*, where *c* is the century code (0 for the 1900s, 1 for 2000–2099), *yy* is the low-order two digits of the year and *ddd* is the day of the year.

STOREPROTECT(*cvda*)

Returns a CVDA value that indicates whether storage protection is active. For storage protection to be active, it must be requested (the STGPROT option in the system initialization table), and the required hardware and software support for storage protection must be available. CVDA values are as follows:

ACTIVE

Storage protection is active.

INACTIVE

Storage protection is not active.

TIME(*data-area*)

Returns a fullword binary field giving the maximum interval in milliseconds for which CICS gives control to the operating system if no tasks are ready for dispatch. This value is set by the ICV option in the system initialization table and is sometimes called the *region exit time interval*.

TRANISOLATE(*cvda*)

Returns a CVDA value that indicates whether transaction isolation is active. For it to be active, both transaction isolation and storage protection must be requested (the TRANISO and STGPROT system initialization parameters), and the required hardware and software support for storage protection must be available. CVDA values are as follows:

ACTIVE

Transaction isolation is active.

INACTIVE

Transaction isolation is not active.

UDSASIZE(*data-area*)

Returns the current size in bytes of the user dynamic storage area (UDSA), in fullword binary form. It includes both storage in use and storage available for use. This size is calculated and managed by CICS automatically, within the DSALIMIT value, that is, the overall limit for dynamic storage areas that reside below 16 MB (below the line).

XRFSTATUS (*cvda*)

Returns a CVDA value that indicates whether the current execution of CICS started as an active or alternate region under the extended recovery facility (XRF):

NOTAPPLIC

CICS is running without XRF support. (XRF=NO in the system initialization table.)

PRIMARY

CICS started as the active region.

TAKEOVER

CICS started as the alternate region.

Conditions**NOTAUTH**

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

INQUIRE TASK

Retrieve information about a user task.

INQUIRE TASK

►►—INQUIRE TASK—(*—data-value—*)—┐ options ┘◄◄

Conditions: INVREQ NOTAUTH TASKIDERR

This command is threadsafe.

Options:

ACTIVITY—(—data-area—)
ACTIVITYID—(—data-area—)
ATTACHTIME—(—data-area—)
BRFACILITY—(—data-area—)
BRIDGE—(—data-area—)
CMDSEC—(—cvda—)
CURRENTPROG—(—data-area—)
DB2PLAN—(—data-area—)
DTIMEOUT—(—data-area—)
DUMPING—(—cvda—)
FACILITY—(—data-area—)
FACILITYTYPE—(—cvda—)
IDENTIFIER—(—data-area—)
INDOUBT—(—cvda—)
INDOUBTMINS—(—data-area—)
INDOUBTWAIT—(—cvda—)
IPFACILITIES—(—ptr-ref—)
IPFLISTSIZE—(—data-area—)
ISOLATEST—(—cvda—)
PRIORITY—(—data-area—)
PROCESS—(—data-area—)
PROCESSTYPE—(—data-area—)
PROFILE—(—data-area—)
PROGRAM—(—data-area—)
PURGEABILITY—(—cvda—)
REMOTENAME—(—data-area—)
REMOTESYSTEM—(—data-area—)
RESSEC—(—cvda—)
ROUTING—(—cvda—)
RTIMEOUT—(—data-area—)
RUNAWAY—(—data-area—)
RUNSTATUS—(—cvda—)
SCRNSIZE—(—cvda—)
STARTCODE—(—data-area—)
STORAGECLEAR—(—cvda—)
SUSPENDTIME—(—data-area—)
SUSPENDTYPE—(—data-area—)
SUSPENDVALUE—(—data-area—)
RESNAME—(—data-area—)
TASKDATAKEY—(—cvda—)
TASKDATALOC—(—cvda—)
TCB—(—cvda—)
TRANCLASS—(—data-area—)
TCLASS—(—data-area—)
TRACING—(—cvda—)
TRANPRIORITY—(—data-area—)
TRANSACTION—(—data-area—)
TRPROF—(—data-area—)
TWASIZE—(—data-area—)
UOW—(—data-area—)
USERID—(—data-area—)

For more information about the user of CVDAs, see CICS-value data areas (CVDA) in CICS System Programming Reference.

Description

The INQUIRE TASK command returns information about a specific user task. User tasks are those associated with user-defined transactions or with CICS-supplied transactions that are normally invoked by an operator.

Many of the options available on this command are the same as those available on the INQUIRE TRANSACTION command, because a task obtains most of its characteristics from the definition of the transaction it is executing. However, these properties are determined at task initiation.

If the transaction definition is changed after the task begins, the task may have a different value for a property than the current transaction definition. Task values can also be changed with a SET TASK command or its CEMT equivalent.

In addition, the INQUIRE TASK command always produces information about the task you specify on the local CICS system. You need to keep this in mind for tasks that are subject to routing or that issue LINK commands that may be shipped to another system.

Whenever a task is executed wholly or in part on a system other than the one on which it originates, there is a matching task on the remote system. The task on the originating system takes its characteristics from the definition on that system of the transaction it is to execute. The corresponding task on the remote system (if routing takes place or the task issues distributed program links) takes its characteristics from the definition of whatever transaction on the remote system that the originating system tells the remote system to use. This remote transaction may have different properties from those of the transaction on the originating system. (It may or may not have a different name; in the case of static transaction routing, the name of the transaction in the remote system comes from the REMOTENAME option of the transaction in the local system.)

Consequently, an inquiry about the task on the originating system may produce entirely different results from an inquiry about the corresponding task on the remote system. For the same reason, a task that issues distributed program links may get a different result from an INQUIRE TASK about itself (taking the task number from the EIB) in a program executing remotely than from the same command in a program executing locally.

Options

ACTIVITY (*data-area*)

Returns the 16-character, user-assigned, name of the BTS activity that this task is executing on behalf of.

ACTIVITYID (*data-area*)

Returns the 52-character, CICS-assigned, identifier of the BTS activity that this task is executing on behalf of.

ATTACHTIME (*data-area*)

Returns an 8-byte packed decimal value, in ABSTIME format, representing the time in milliseconds at which the task was attached.

BRFACILITY (*data-area*)

Returns the 8-byte facility token representing the virtual terminal used by the current task if it is running in a bridged environment. If the task is not running in the 3270 bridge environment, zeroes are returned.

BRIDGE (*data-area*)

Returns the 4-character transaction identifier of the bridge monitor transaction that issued a START BREXIT TRANSID command to start this task, or the client that issued a link to DFHL3270. If the task is not currently running in the 3270 bridge environment, then blanks are returned.

CMDSEC (*cvda*)

Returns a CVDA value indicating whether the definition of the transaction the task is executing specifies command security. CVDA values are as follows:

CMDSECNO

Command security is not specified.

CMDSECYES

Command security is specified.

When a task being checked issues a system programming command, CICS calls the external security manager (ESM) to verify that the user associated with the task has authority to use these commands.

A task is command-checked only when an ESM is active and either the CMDSEC value for the task is CMDSECYES or the system initialization option CMDSEC value is ALWAYS (see the SECURITYMGR option of INQUIRE SYSTEM in CICS System Programming Reference for more information).

CURRENTPROG (*data-area*)

Returns a 1- to 8-character name of the current program, as known to the CICS program manager domain, executing for this task.

DB2PLAN (*data-area*)

Returns a 1- to 8-character name of the DB2PLAN being used by this task, or blanks if no DB2PLAN is being used.

DTIMEOUT (*data-area*)

Returns a fullword binary field giving the deadlock timeout interval, in seconds. CICS abends a task that waits longer than its deadlock timeout value for a locked resource.

DUMPING (*cvda*)

Returns a CVDA value indicating whether CICS should take a transaction dump if the task terminates abnormally. CVDA values are as follows:

NOTRANDUMP

No dump is taken.

TRANDUMP

A dump is taken.

This value applies only to abend dumps and has no effect on DUMP TRANSACTION commands.

FACILITY (*data-area*)

Returns the 4-character name of the facility associated with initiation of this task, if that facility is a transient data queue or a terminal or system. If the task was initiated otherwise, the facility value is blanks. The FACILITYTYPE option tells you what type of facility caused task initiation, and therefore what FACILITY represents.

FACILITYTYPE (*cvda*)

Returns a CVDA value identifying the type of facility that initiated this task. CVDA values are:

DEST CICS initiated the task to process a transient data queue which had reached trigger level; the FACILITY option returns the name of queue.

TASK Another task initiated the task with a START command that did not specify a terminal, or CICS created the task internally; the FACILITY option returns blanks in this case.

TERM Either the task was initiated to process unsolicited input or another task initiated the task with a START command with the TERMDID option. In the first case the FACILITY option returns the name of the terminal that sent the input, and in the second, it returns the terminal named in TERMDID.

IDENTIFIER(*data-area*)

Returns a 48-character field containing user data provided by the bridge exit, if the task was initiated in the 3270 bridge environment, or blanks, otherwise. This field is intended to assist in online problem resolution. For example, it could contain the WebSphere MQ correlator for the CICS-WebSphere MQ bridge, or a Web token.

INDOUBT(*cvda*)

Returns a CVDA value, based on the ACTION attribute of the TRANSACTION resource definition, indicating the action to be taken if the CICS region fails, or loses connectivity with its coordinator while a unit of work is in the indoubt period.

The action is dependent on the values returned in INDOUBTWAIT and INDOUBTMINS; if INDOUBTWAIT returns WAIT, the action is not taken until the time returned in INDOUBTMINS expires.

CVDA values are:

BACKOUT

All changes made to recoverable resources are to be backed out.

COMMIT

All changes made to recoverable resources are to be committed, and the unit of work marked as completed.

Note: If a program uses the obsolete DTB option, which was replaced by INDOUBT, a CVDA value of NOTSUPPORTED is returned.

INDOUBTMINS(*data-area*)

Returns a fullword binary field giving the length of time, in minutes, after a failure during the indoubt period, before the task is to take the action returned in the INDOUBT field. The returned value is valid only if the unit of work is indoubt and INDOUBTWAIT returns WAIT.

See also INDOUBT and INDOUBTWAIT.

INDOUBTWAIT(*cvda*)

Returns a CVDA value, based on the WAIT attribute of the TRANSACTION definition, indicating how a unit of work (UOW) is to respond if a failure occurs while it is in an indoubt state. CVDA values are:

NOWAIT

The unit of work is not to wait, pending recovery from the failure. CICS is to take immediately whatever action is specified on the ACTION attribute of the TRANSACTION definition.

WAIT The unit of work is to wait, pending recovery from the failure, to determine whether recoverable resources are to be backed out or committed.

For further information about the meaning of the ACTION and WAIT attributes of the TRANSACTION definition, see TRANSACTION definition attributes.

IPFACILITIES(*ptr-ref*)

Returns the address of a list of 4-byte binary tokens, each of which identifies an IPCONN session that the task is using to communicate with another system. If there are no such IP facilities for this task, the IPFACILITIES pointer contains a null value.

CICS obtains the storage for the list and frees it when the inquiring task issues another INQUIRE TASK command or ends; the task cannot free the storage itself.

IPFLISTSIZE(*data-area*)

Returns a fullword binary field giving the number of IP facilities associated with this task. (That is, it returns the number of items in the list addressed by the IPFACILITIES option.)

If this task has no IP facilities, IPFLISTSIZE contains zero.

ISOLATEST(*cvda*)

Returns a CVDA value indicating whether the task is defined as isolated or not. Isolation limits the access, for both read and write, of user-key programs to task storage. A program executing in user key on behalf of an isolated task can access the task storage of only that task, and this storage cannot be accessed by programs executing in user key on behalf of other tasks. Isolation does not affect access by CICS-key programs and does not apply to storage with the SHARED attribute or any other nontask storage.

The value of ISOLATEST is taken from the definition of the TRANSACTION the task is executing when the task is created. For a task defined as isolated to execute isolated, transaction isolation for the system must also be active (see the TRANISOLATE option of INQUIRE SYSTEM in CICS System Programming Reference).

ISOLATE

The task is defined as isolated.

NOISOLATE

The task is defined as not isolated.

PRIORITY(*data-area*)

Returns a fullword binary field giving the total priority of the task. Total priority is the sum of the priority of the user associated with the task, the priority of the terminal which is the principal facility, and the priority of the transaction being executed (see the TRANPRIORITY option).

PROCESS(*data-area*)

Returns the 36-character name of the BTS process of which this task is a part.

PROCESSTYPE(*data-area*)

Returns the 8-character identifier of the type definition of the BTS process of which this task is a part.

PROFILE(*data-area*)

Returns the 8-character name of the PROFILE for the transaction this task is executing.

PROGRAM(*data-area*)

Returns the 8-character name of the program executed first in this task.

PURGEABILITY(*cvda*)

Returns a CVDA value indicating whether CICS is allowed to purge this task (that is, to terminate it abnormally). Purge requests come from SET TASK PURGE commands (or CEMT equivalents), and CICS can generate them internally to reclaim resources to relieve a system stall condition. CVDA values are:

NOTPURGEABLE

The task cannot be purged.

PURGEABLE

The task can be purged.

The PURGEABILITY value is set initially by the SPURGE option in the definition of the transaction this task is executing.

REMOTENAME(*data-area*)

Returns the 4-character name assigned in the REMOTENAME option of the definition of the TRANSACTION which this task is executing. When CICS routes a task statically, REMOTENAME is the name of the transaction that the partner task on the remote system executes. Consequently REMOTENAME is significant to the task about which you are inquiring only if it is subject to routing.

CICS returns blanks if the transaction definition does not specify REMOTENAME.

REMOTESYSTEM(*data-area*)

Returns the 4-character name assigned in the REMOTESYSTEM option of the definition of the TRANSACTION which this task is executing. When CICS routes a task statically, REMOTESYSTEM is the name of the CONNECTION definition of the system to which the task is routed. Like REMOTENAME, REMOTESYSTEM is significant to the task about which you are inquiring only if it is subject to routing.

CICS returns blanks if the TRANSACTION definition does not specify REMOTESYSTEM.

RESNAME(*data-area*)

RESNAME, an alternative to SUSPENDVALUE, returns a 16-character resource name of tasks suspended on TS queues.

RESSEC(*cvda*)

Returns a CVDA value indicating whether the definition of the TRANSACTION the task is executing specifies resource-level security checking. CVDA values are:

RESSECNO

Command security is not specified.

RESSECYES

Command security is specified.

When a task is being checked, CICS verifies on each command that the user associated with the task has authority to access the resource named in the way requested.

A task is checked only when an external security manager is active and either the RESSEC value for the task is RESSECYES or the system initialization

option RESSEC value is ALWAYS (see the SECURITYMGR option of INQUIRE SYSTEM in CICS System Programming Reference for more information.

ROUTING(*cvda*)

Returns a CVDA value indicating whether the transaction this task is executing specifies dynamic routing or not (in the DYNAMIC option in the TRANSACTION definition). Dynamic routing occurs just before the initial dispatch of a task, and therefore this value indicates whether dynamic routing may have occurred (if the task is already in execution) or may yet occur (if it has not yet been dispatched). CVDA values are:

DYNAMIC

Dynamic routing applies.

STATIC

Dynamic routing does not apply.

RTIMEOUT(*data-area*)

Returns a fullword binary field giving the read timeout interval, in seconds. CICS abends a task if it waits for input longer than its read timeout value. The RTIMEOUT value is set by the RTIMOUT option in the PROFILE definition associated with the TRANSACTION this task is executing.

RUNAWAY(*data-area*)

Returns the "runaway task" time for this task, in milliseconds, as a fullword binary value. If a task keeps control of the processor for more than this interval on a single dispatch, CICS assumes it is in a loop and abends it. If the value is zero, CICS does not monitor the task for a runaway condition.

RUNSTATUS(*cvda*)

Returns a CVDA value indicating the dispatch status of the task. CVDA values are:

DISPATCHABLE

The task is ready to run.

RUNNING

The task is running.

SUSPENDED

The task is not ready to run.

SCRNSIZE(*cvda*)

Returns a CVDA value indicating whether the alternate or the default screen size applies to this task. CVDA values are:

ALTERNATE

Alternate screen size applies.

DEFAULT

Default screen size applies.

The SCRNSIZE value is set by the same-named option in the PROFILE definition associated with the transaction this task is executing.

STARTCODE(*data-area*)

Returns a 2-character value indicating how this task started. Possible values are:

D The task was initiated to process a distributed programming link (DPL) command that did not specify the SYNCONRETURN option. (The task is not allowed to issue syncpoints.)

- DS** The task was initiated to process a distributed programming link (DPL) command containing the SYNCONRETURN option. (The task is allowed to issue syncpoints).
- QD** CICS initiated the task to process a transient data queue that had reached trigger level.
- S** Another task initiated this one, using a START command that did not pass data in the FROM option. The START command may or may not have passed a channel.
- SD** Another task initiated this one, using a START command that passed data in the FROM option.
- SZ** The task was initiated with a FEPI START command.
- TO** The task was initiated to process unsolicited input from a terminal (or another system), and the transaction to be executed was determined from the input.
- TP** The task was initiated to process unsolicited input or in response to a RETURN IMMEDIATE command in another task. In either case, the transaction to be executed was preset (in the RETURN command or in the associated TERMINAL definition) without reference to input.
- U** CICS created the task internally.

Note: When the IOP request processor is run locally the startcode for an ASSIGN command or an INQUIRE TASK is U. When the IOP request processor is run remotely, over an MRO link, the startcode for these commands is TO. (If you attempt to run the IOP request processor remotely over any other type of connection, the routing request is not accepted, so startcodes for these commands are not relevant in this situation).

STORAGECLEAR(*cvda*)

Returns a CVDA value indicating whether CICS should clear storage that is released from this task (to prevent other tasks accidentally viewing confidential data). CVDA values are:

CLEAR

Storage is cleared.

NOCLEAR

Storage will not be cleared.

SUSPENDTIME(*data-area*)

Returns a fullword binary field giving the number of seconds (rounded down) for which the task has been suspended since last dispatch, if its RUNSTATUS value is SUSPENDED. If the task is running or dispatchable, the SUSPENDTIME value is -1 .

SUSPENDTYPE(*data-area*)

Returns an 8-character text string indicating why this task is suspended, if it is (blanks are returned for tasks that are running or dispatchable). See the SUSPENDVALUE option also.

SUSPENDVALUE(*data-area*)

Returns the 8-character name of the resource for which this task is waiting (the name of the file if the task is enqueued on a record, for example). SUSPENDVALUE applies only to suspended tasks; if the task is running or dispatchable, the value returned is blanks.

For information on the values that can appear in the SUSPENDTYPE and SUSPENDVALUE options, and how they can be used as an aid in problem determination, see the resource type and resource name details in the *CICS Problem Determination Guide*.

TASK(*data-value*)

Specifies the 4-byte packed-decimal sequence number of the task to be inquired upon.

TASKDATAKEY(*cvda*)

Returns a CVDA value indicating the storage key in which CICS obtains storage for this task. This includes the task life-time storage - the transaction work area (TWA) and the EXEC interface block (EIB) - and the storage that CICS obtains on behalf of programs that run under this task.

See the description of the TASKDATAKEY option in TRANSACTION definition attributes for more information.

CVDA values are as follows:

CICSDATAKEY

CICS obtains storage from CICS-key storage.

USERDATAKEY

CICS obtains storage from user-key storage.

The value returned for TASKDATAKEY is taken from the definition of the TRANSACTION that the task is executing. To determine whether storage protection is active (that is, whether user-key has a different value from CICS-key), you need to issue an INQUIRE SYSTEM command with the STOREPROTECT option.

TASKDATALOC(*cvda*)

Returns a CVDA value indicating whether task-lifetime storage for this task (CICS control blocks for the task such as the EIB and TWA) should be acquired above or below the 16 MB line. CVDA values are as follows:

ANY Task-lifetime storage can be either below or above the 16 MB line.

BELOW

Task-lifetime storage must be below the 16 MB line.

TCB(*cvda*)

Returns a CVDA value indicating the type of TCB under which the task is running. The CVDA values are as follows:

CKOPEN

The task is running under a CICS key open TCB (for example, a J8 mode TCB, which is used for a JVM in CICS key).

INTERNAL

The task is running under one of the CICS internal TCBs. An internal TCB can be one of the following:

- The concurrent mode (CO) TCB
- The file-owning mode (FO) TCB
- The shared class cache (JM) TCB
- The resource-owning mode (RO) TCB
- The ONC/RPC mode (RP) TCB
- The sockets listener mode (SL) TCB
- The secure sockets layer mode (SO) TCB

- A sockets mode (S8) TCB
- The FEPI mode (SZ) TCB.

QR The task is running under the CICS QR TCB.

UKOPEN

The task is running under a user key open TCB (for example, a J9 mode TCB, which is used for a JVM in user key).

TCLASS(*data-area*)

Returns a fullword binary field giving the number of the transaction class to which this task belongs, if it belongs to a numbered transaction class. This option is retained for compatibility with earlier releases, where transaction classes were numbered from 1 to 10. If the task does not belong to such a class, the value returned is zero. (See the TRANCLASS option for more information.)

TRACING(*cvda*)

Returns a CVDA value indicating the type of tracing in effect for this task. CVDA values are:

SPECTRACE

Tracing for this task is special.

SPRSTRACE

Tracing for this task is suppressed.

STANTRACE

Tracing for this task is standard.

For further information on the types of tracing, see *CICS Problem Determination Guide* and *CICS Supplied Transactions*.

TRANCLASS(*data-area*)

Returns the 8-character name of the transaction class to which the task belongs. If the task is not assigned to any class, the default class DFHTCL00 is returned. If the task belongs to a numbered class, the value returned is DFHTCL nn , where nn is the 2-digit class number.

TRANPRIORITY(*data-area*)

Returns a fullword binary field giving the component of the total priority of the task that came from the PRIORITY option in the definition of the TRANSACTION being executed. (See the PRIORITY option of this command also.)

TRANSACTION(*data-area*)

Returns the 4-character name of the transaction that this task is executing.

TRPROF(*data-area*)

Returns the 8-character name of the profile definition used for intersystem flows if the task is routed on an ISC link.

TWASIZE(*data-area*)

Returns a fullword binary field giving the size in bytes of the transaction work area (TWA) for this task.

UOW(*data-area*)

Returns, as an 8-byte field, the local identifier of the unit of work associated with this task.

USERID(*data-area*)

Returns the 8-character identifier of the user associated with the task.

Conditions

INVREQ

RESP2 values:

- 1 SUSPENDVALUE is specified, but significant characters are lost.
- 3 TCLASS is specified, but the task belongs to a named CLASS, not a numbered CLASS. The user should specify the TRANCLASS option.
- 10 The requested data is held on a data profile, but the data profile is not available.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

TASKIDERR

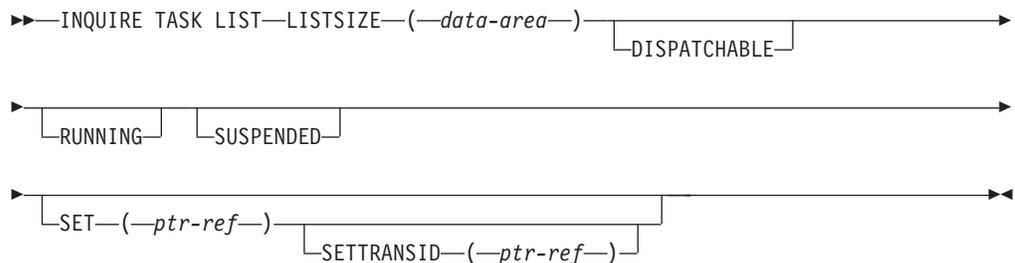
RESP2 values:

- 1 The task cannot be found.
- 2 The task is executing a type of transaction which is not subject to this command.

INQUIRE TASK LIST

Retrieve a list of user tasks.

INQUIRE TASK LIST



Condition: NOTAUTH

Description

The INQUIRE TASK LIST command returns a list of user tasks. User tasks are tasks associated with user-defined transactions or with CICS-supplied transactions that are normally invoked by an operator. You can restrict the list to tasks that are DISPATCHABLE (ready to run), RUNNING, or SUSPENDED at the time of the inquiry, or any combination of these.

Options

DISPATCHABLE

specifies that tasks ready to run (dispatchable) should be included in the task list. These tasks are also included if you specify none of the category options (DISPATCHABLE, RUNNING, and SUSPENDED).

LISTSIZE(*data-area*)

returns a fullword binary field giving the number of tasks in the categories you included in your inquiry. This is the number of entries in the lists that the SET and SETTRANSID options produce. If there are no tasks in the categories requested, LISTSIZE contains zero.

RUNNING

specifies that the tasks executing (including the one issuing the command) should be included in the task list. The tasks are also included if you specify none of the category options (DISPATCHABLE, RUNNING, and SUSPENDED).

SET(*ptr-ref*)

returns the address of a list of 4-byte packed-decimal task numbers. Each entry in the list identifies a task in one of the categories requested (see the DISPATCHABLE, RUNNING, and REQUESTED options). If there are no tasks in the categories requested, the SET pointer contains a null value.

CICS obtains the storage for this list and frees it when the inquiring task issues another INQUIRE TASK LIST, or issues an INQUIRE STORAGE command with ELEMENTLIST or LENGTHLIST, or ends; the task cannot free the storage itself.

SETTRANSID(*ptr-ref*)

returns the address of a list of 4-byte transaction identifiers. Each entry in the list is the name of the transaction that the task in the corresponding entry in the SET list is executing. If there are no tasks in the categories that you have specified, the SETTRANSID pointer contains a null value.

CICS obtains the storage for this list and frees it when the inquiring task issues another INQUIRE TASK LIST, or issues an INQUIRE STORAGE command with ELEMENTLIST or LENGTHLIST, or ends; the task cannot free the storage itself.

SUSPENDED

specifies that suspended tasks (tasks waiting for some event or condition) should be included in the task list. For this purpose, tasks which have not reached the point of initial dispatch, either because the task class to which they belong is at its maximum or because the maximum for the system has been reached, are considered suspended. Suspended tasks are also included if you specify none of the category options (DISPATCHABLE, RUNNING, and SUSPENDED).

Conditions**NOTAUTH**

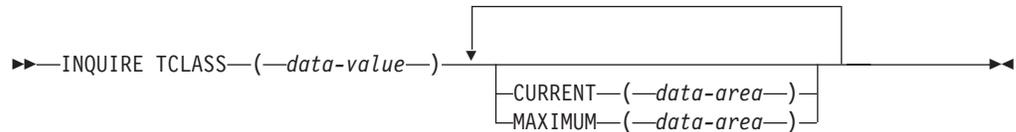
RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

INQUIRE TCLASS

Retrieve information about a transaction class.

INQUIRE TCLASS



Conditions: NOTAUTH, TCIDERR

Description

Use the **INQUIRE TCLASS** command to determine the current and maximum numbers of tasks in an installation-defined transaction class. This command is limited to the numbered classes of earlier releases of CICS and is retained for compatibility with those releases. The "INQUIRE TRANCLASS" on page 511 command has the same function and can be used for either the old numbered or the new named classes.

Options

CURRENT (*data-area*)

Returns a fullword binary field giving the current number of tasks in the class about which you are inquiring. This number includes both tasks that are running and tasks that have not yet been dispatched because the maximum for either the class or the system has been reached. See the **MAXIMUM** option of this command and the **MAXTASKS** option of the **INQUIRE SYSTEM** command for more about these limits. The **CURRENT** value corresponds to the sum of the **ACTIVE** and **QUEUED** values in an **INQUIRE TRANCLASS** command, and therefore can exceed the **MAXIMUM** value.

MAXIMUM (*data-area*)

Returns a fullword binary field giving the largest number of tasks that are allowed to run concurrently in the class about which you are inquiring. This value corresponds to the **MAXACTIVE** value in an **INQUIRE TRANCLASS** command.

TCLASS (*data-value*)

Specifies the number of the task class about which you are inquiring, in fullword binary form. The number must be in the range 0-10.

Conditions

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

TCIDERR

RESP2 values:

1 The named task class cannot be found.

INQUIRE TCPIP

Retrieve information about CICS internal sockets support.

INQUIRE TCPIP

▶▶ INQUIRE TCPIP ACTSOCKETS—(*data-value*)—————▶
▶ MAXSOCKETS—(*data-value*)—OPENSTATUS—(*cvda*)—SSLCACHE—(*cvda*)————▶
▶ CRLPROFILE—(*data-value*)—————▶▶

Conditions: INVREQ, NOTAUTH

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

INQUIRE TCPIP returns information about the state of CICS internal sockets support.

Options

ACTSOCKETS (*data-value*)

Returns a fullword binary field containing the current number of active IP sockets managed by the CICS sockets domain.

CRLPROFILE (*data-value*)

Returns the name of the profile that authorizes CICS to access the LDAP server that stores certificate revocation lists for SSL connections.

MAXSOCKETS (*data-value*)

Returns a fullword binary field containing the maximum number of IP sockets that can be managed by the CICS sockets domain.

OPENSTATUS (*cvda*)

Returns a CVDA value indicating the status of CICS internal sockets support. CVDA values are:

OPEN

CICS internal TCPIP support is open.

CLOSED

CICS internal sockets support has not yet been activated, or has been terminated.

CLOSING

CICS internal sockets support is in the process of closing.

IMMCLOSING

CICS internal sockets support is in the process of immediate termination.

SSLCACHE (*cvda*)

Returns a CVDA value indicating if CICS is configured to use local or sysplex caching for SSL session ids. CVDA values are:

CICS CICS is configured to cache SSL session ids in the local CICS region.

SYSPLEX

CICS is configured to cache SSL session ids in the coupling facility.

Conditions

INVREQ

RESP2 values:

- 4 TCPIP=NO has been specified in the system initialization table.

NOTAUTH

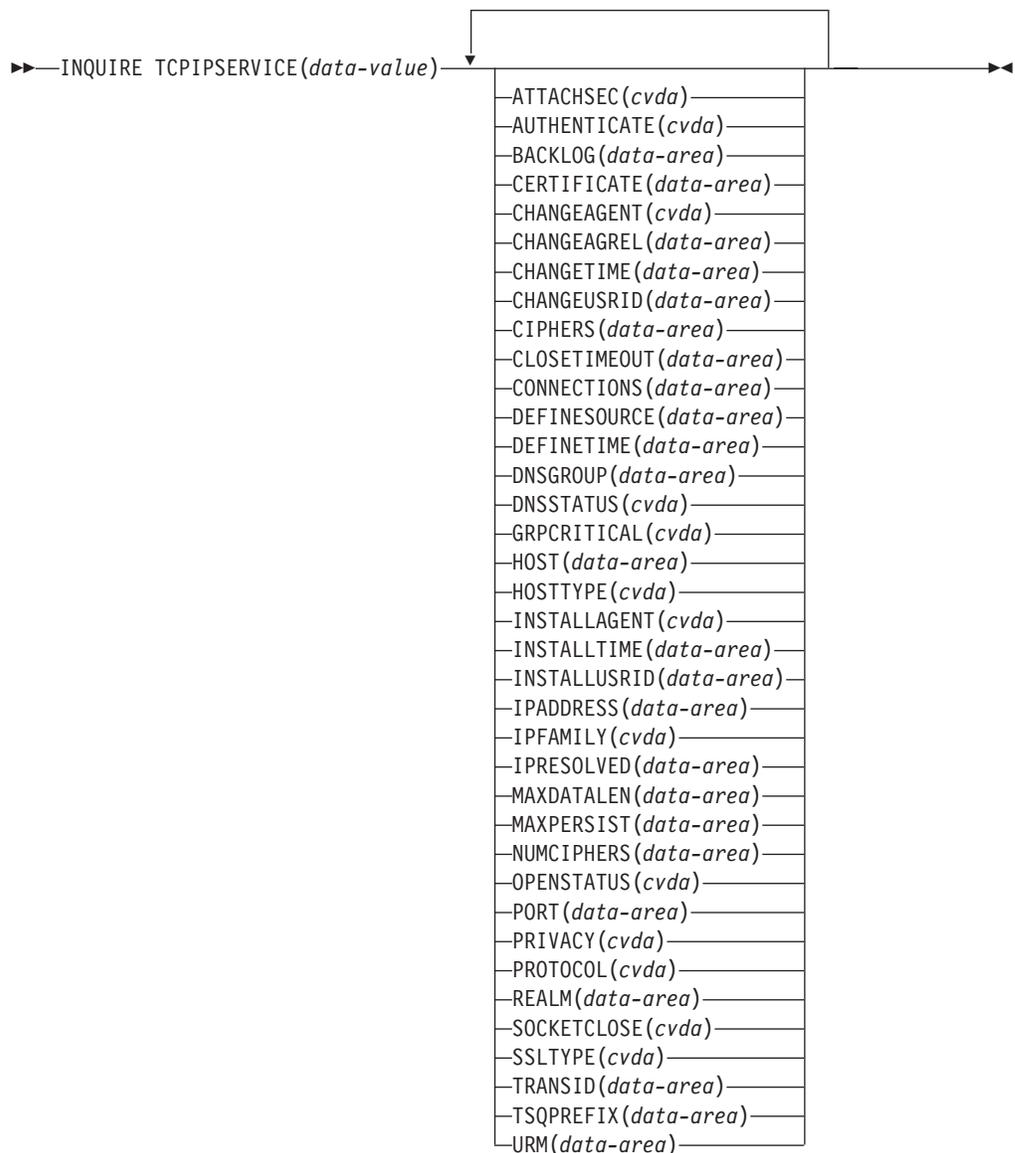
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

INQUIRE TCIPSERVICE

Retrieve information about the state of a service by using CICS internal TCP/IP support.

INQUIRE TCIPSERVICE



Conditions: END, ILLOGIC, INVREQ, NOTAUTH, NOTFND

For more information about the use of CVDAs, see CICS-value data areas (CVDA) in CICS System Programming Reference.

Description

Use the INQUIRE TCPIP SERVICE command to retrieve information about TCP/IP ports on which CICS internal TCP/IP support is currently listening on behalf of other CICS services.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

ATTACHSEC(*cvda*)

Returns, for ECI over TCP/IP and IPIC services, the level of attach-time user security used by the connection. CVDA values are as follows:

LOCAL

CICS does not require a user ID or password from clients.

VERIFY

Incoming attach requests must specify a user ID and a user password.

This option has no meaning for CICS web support or IIOP TCP/IP connections.

AUTHENTICATE(*cvda*)

Returns a CVDA indicating the scheme used to authenticate clients. Possible values are as follows:

- ASSERTED
- AUTOAUTH
- AUTOREGISTER
- BASICAUTH
- CERTIFICAUTH
- NOAUTHENTIC

ASSERTED

Asserted identity authentication is used to authenticate the client. This value is returned only when PROTOCOL has a value of IIOP.

AUTOAUTH

If the client does not send a certificate, HTTP basic authentication is used to obtain a user ID and password from the client. Otherwise, SSL client certificate authentication is used to authenticate the client. If the

client certificate is not associated with a user ID, HTTP basic authentication is used to obtain the client user ID, and associate it with the certificate.

This value is returned only when PROTOCOL has a value of HTTP.

AUTOREGISTER

SSL client certificate authentication is used to authenticate the client. If the client certificate is not associated with a user ID, then HTTP basic authentication is used to obtain the client user ID and associate it with the certificate.

This value is returned only when PROTOCOL has a value of HTTP.

BASICAUTH

HTTP basic authentication is used to obtain a user ID and password from the client.

This value is returned only when PROTOCOL has a value of HTTP.

CERTIFICAUTH

SSL client certificate authentication is used to authenticate and identify the client.

This value is returned only when PROTOCOL has a value of HTTP or IIOF.

NOAUTHENTIC

The client is not required to send authentication or identification information. However, if the client sends a valid certificate that is already registered to the security manager, and associated with a user ID, that user ID identifies the client.

This value is returned only when PROTOCOL has a value of HTTP or IIOF.

For more information about authentication and identification of HTTP and IIOF clients, see the *CICS RACF Security Guide*.

BACKLOG(*data-area*)

Returns, in fullword binary form, the maximum number of requests that can be queued in TCP/IP waiting to be processed by the service.

CERTIFICATE(*data-area*)

Returns a 32-character area containing the label of the certificate, in the key ring, that is used as the server certificate in the SSL handshake for all secure socket layer connections on this service.

CHANGEAGENT(*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

SYSTEM

The resource definition was last changed by the CICS or CICSplex system.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

CIPHERS (*data-area*)

Returns a 56-character area containing the list of cipher suites that is used to negotiate with clients during the SSL handshake. The list is set by the **ENCRYPTION** system initialization parameter, but you can edit the list to remove or change the order of cipher suites. See the *CICS RACF Security Guide*.

CLOSETIMEOUT (*data-area*)

Returns, in fullword binary form, the number of seconds that have been specified for this service to wait for data for a new request. This number can be 0 - 86400 (24 hours). For the HTTP protocol, do not specify 0, because this setting means that persistent connections cannot be maintained.

CONNECTIONS

Returns, in fullword binary form, the number of sockets connections for this service.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT** value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

DNSGROUP (*data-area*)

Returns the 18-character DNS group name that CICS registers with the z/OS Workload Manager (WLM) for this TCPIP SERVICE resource.

DNSSTATUS (*cvda*)

Returns the current state of WLM/DNS registration of this TCPIP SERVICE resource. The CVDA values are as follows:

NOTAPPLIC

This service is not using DNS connection optimization. No **DNSGROUP** attribute was specified when the resource was installed.

UNAVAILABLE

Registration is not supported by z/OS.

UNREGISTERED

Registration has not yet occurred. UNREGISTERED is the initial state of any service.

REGISTERED

Registration has completed successfully.

REGERROR

Registration has failed with an error.

DEREGISTERED

Deregistration has completed successfully.

DEREGERROR

Deregistration has failed with an error.

GRPCRITICAL (*cvda*)

Returns a CVDA specifying whether this TCPIPSERVICE resource is a critical member of the DNS group. The CVDA values are as follows:

CRITICAL

If this TCPIPSERVICE resource is closed, or abnormally stops listening for any reason, the group name specified in the DNSGROUP attribute is deregistered from WLM.

NONCRITICAL

If this TCPIPSERVICE resource is closed, or abnormally stops listening for any reason, the group name specified in the DNSGROUP attribute is not deregistered from WLM, unless this service is the last in a set with the same group name.

HOST (*data-area*)

Returns the 116-character host name of the remote system or its IP address.

HOST displays character host name, an IPv4 address, an IPv6 address, ANY, or DEFAULT. The HOST option provides the same function as IPADDRESS for defined hostnames and defined IPv4 addresses, but also supports defined IPv6 format addresses. However, it differs from IPADDRESS in that DEFAULT and ANY are returned instead of an IP address, because this information is available in IPRESOLVED. If you are using IPv6 connections, use the HOST option for your queries, instead of IPADDRESS. HOST displays all IPv4 addresses as native IPv4 dotted decimal addresses; for example, 1.2.3.4, regardless of the type of address format used.

You can specify IPv4 and IPv6 addresses in a number of acceptable formats. See IP addresses in the Internet Guide for more information about address formats.

HOST is specified in the resource definition.

HOSTTYPE (*cvda*)

Returns the address format of HOST, or if HOST is not specified the IPADDRESS option. HOSTTYPE is set by the domain when the TCPIPSERVICE is installed. The CVDA values are as follows:

ANY The ANY option is specified for the HOST option.

DEFAULT

The DEFAULT option is specified for the HOST option.

HOSTNAME

The HOST option contains a character host name. The IP address that corresponds to the host name is looked up in the domain name server.

IPV4 The HOST option contains a dotted decimal IPv4 address.

IPV6 The HOST option contains a colon hexadecimal IPv6 address.

NOTAPPLIC

0.0.0.0 is specified in the HOST option.

INSTALLAGENT(*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

IPADDRESS(*data-area*)

Returns the 15-character dotted decimal IP address of this service. Do not use IPADDRESS for new programs; use HOST instead. The HOST option returns the same information as IPADDRESS, but can also return an IPv6 format address. If HOST returns an IPv4 address, this address is also returned to IPADDRESS; otherwise, IPADDRESS returns 0.0.0.0.

If you are using IPv6 connections, you must use the HOST option for your queries, instead of IPADDRESS.

IPFAMILY(*cvda*)

Returns the address format of the IPRESOLVED option. The CVDA values are as follows:

UNKNOWN

IPRESOLVED is not yet used or the address cannot be resolved.
UNKNOWN is the default when IPRESOLVED is 0.0.0.0.

IPV4 The IPRESOLVED option contains a dotted decimal IPv4 address.

IPV6 The IPRESOLVED option contains a colon hexadecimal IPv6 address.

IPRESOLVED(*data-area*)

Returns, in a 39-character area, the IPv4, or IPv6 address of the HOST option. If the OPENSTATUS option is not set to OPEN, or the address cannot be resolved, a value of 0.0.0.0 is returned. If the HOST option is set to ANY, IPRESOLVED always returns the IPv4 address for the system on which CICS is running, even if other IPv4 or IPv6 addresses are available.

The content of IPRESOLVED is not recoverable after a warm or emergency restart.

MAXDATALEN(*data-area*)

Returns, in fullword binary form, the setting for the maximum length of data that can be received by CICS as an HTTP server.

| **MAXPERSIST** (*data-area*)

| Returns, in fullword binary form, the setting for the maximum number of
| persistent connections from web clients that the CICS region allows for this
| port at any one time. This setting applies only for the HTTP protocol. A null
| setting (-1) means that there is no limit on the number of persistent
| connections. A zero setting means that no persistent connections are allowed. A
| zero setting is not compliant with the HTTP/1.1 specification and must not be
| set in a CICS region that is handling external requests.

| **NUMCIPHERS** (*data-area*)

| Returns a binary halfword data area that contains the number of cipher suites
| that are specified in the CIPHERS attribute.

| **OPENSTATUS** (*cvda*)

| Returns a CVDA indicating the status of CICS internal sockets support for the
| service. CVDA values are as follows:

| **OPEN**

| CICS internal sockets support is open for this service.

| **OPENING**

| CICS internal sockets support is in the process of opening for this
| service.

| **CLOSED**

| CICS internal sockets support has not yet been activated, or has been
| ended, for this service.

| **CLOSING**

| CICS internal sockets support is in the process of closing for this
| service.

| **IMMCLOSE**

| CICS internal sockets support has immediately terminated for this
| service.

| **IMMCLOSING**

| CICS internal sockets support is in the process of immediate
| termination.

| **PORT** (*data-area*)

| Returns, in fullword binary form, the number of the port on which CICS is
| listening on behalf of this service.

| **PRIVACY** (*cvda*)

| Returns a CVDA indicating the level of SSL encryption required for inbound
| connections to this service. CVDA values are as follows:

| **REQUIRED**

| Encryption must be used. During the SSL handshake, CICS advertises
| only supported cipher suites that provide encryption.

| **SUPPORTED**

| Encryption is used if both client and server support it. During the SSL
| handshake, CICS advertises all supported cipher suites.

| **NOTSUPPORTED**

| Encryption must not be used. During the SSL handshake, CICS
| advertises only supported cipher suites that do not provide encryption.

| **PROTOCOL** (*cvda*)

| Returns a CVDA indicating the underlying protocol being used on this service.
| CVDA values are as follows:

ECI External CICS interface protocol.

HTTP Hypertext Transfer protocol.

IIOP Internet Inter-Orb protocol.

IPIC IP interconnectivity (IPIC).

USER User-defined protocol.

REALM(*data-area*)

Returns the 56-character realm that is used during the process of HTTP basic authentication. This value is returned only when PROTOCOL has a value of HTTP. If no realm is specified for this service, the default realm used by CICS is returned, which is CICS application *aaaaaaaa*, where *aaaaaaaa* is the APPLID of the CICS region.

SOCKETCLOSE(*cvda*)

Returns a CVDA telling you whether a TIMEOUT value is in effect for this service. CVDA values are as follows:

WAIT NO was specified on the definition. Socket receives wait for data indefinitely.

TIMEOUT

A value was specified for the SOCKETCLOSE parameter on the definition. CLOSETIMEOUT returns the specified value.

SSLTYPE(*cvda*)

Returns a CVDA specifying the level of secure sockets support being used for this service. CVDA values are as follows:

CLIENTAUTH

The Secure Sockets Layer with client authentication is being used for this service.

NOSSL

The Secure Sockets Layer is not being used for this service.

SSL The Secure Sockets Layer without client authentication is being used for this service.

TCPIPSERVICE(*data-value*)

Specifies the 1- to 8-character name of the TCP/IP service about which you are inquiring.

TRANSID(*data-area*)

Returns the 4-character transaction ID used on the attach for the task started to process a new request.

TSQPREFIX(*data-area*)

Is no longer required or used in CICS Transaction Server for z/OS, Version 3 Release 2 and later releases.

URM(*data-area*)

Returns the 8-character name of the user-replaceable program to be started by the attached task.

Conditions

END

RESP2 values:

2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

INVREQ

RESP2 values:

- 4 TCP/IP not available (TCPIP=NO was specified as a system initialization parameter)
- 5 TCP/IP is closed.
- 19 Unknown host.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

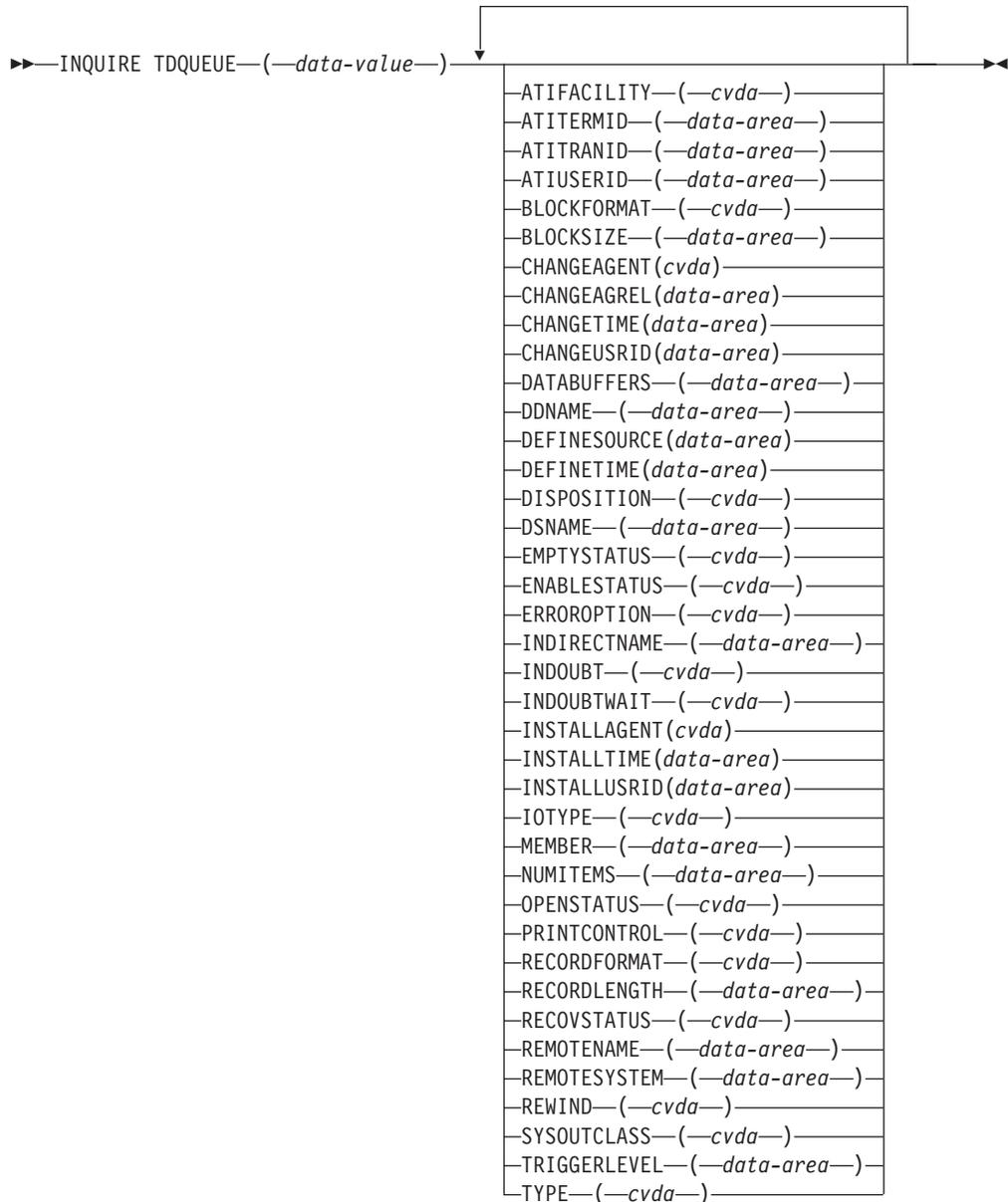
RESP2 values:

- 3 The TCPIPSERVICE resource was not found.

INQUIRE TDQUEUE

Retrieve information about a transient data queue.

INQUIRE TDQUEUE



Conditions: ENDCOND, ILLOGIC, NORMAL, NOTAUTH, QIDERR

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The INQUIRE TDQUEUE command retrieves information about a particular transient data queue.

You define transient data queues to CICS using transient data resource definitions. There are two basic types: *intrapartition* and *extrapartition*. Intrapartition queues are managed and stored entirely by CICS, and are subject to automatic task initiation

(ATI). ATI means that when the number of items on the queue reaches the value in the TRIGGERLEVEL option, CICS automatically creates a task to process the queue.

An extrapartition queue is an MVS sequential data set or a spool file. Extrapartition queues are not subject to ATI, and consequently the associated options produce null values. Furthermore, if the data set is not open, CICS might not be able to determine some of the values, such as BLOCKFORMAT and RECORDFORMAT. Null values, explained in “Null values” on page 14, are returned in such cases.

Two other types of queue exist: *indirect* and *remote*, both of which point, eventually, to one of the basic types.

An indirect queue points to another queue on the same CICS system, and is essentially an alias for the other queue. When you name an indirect queue in an INQUIRE TDQUEUE command, CICS returns only the TYPE value, which is INDIRECT, and the name of the queue to which the indirect definition points (the INDIRECTNAME value). You need a second INQUIRE TDQUEUE against the INDIRECTNAME value to determine the characteristics of the underlying queue.

A remote queue is one defined on another CICS system. When you inquire about such a queue, the local CICS system returns only the information it maintains locally about the queue:

- The TYPE (REMOTE).
- The system on which it is defined (the REMOTESYSTEM value).
- Its name there (REMOTENAME).
- Whether it is available to applications on the local system (its ENABLESTATUS).

Browsing

You can also browse through the transient data queues defined in your system by using the browse options, START, NEXT, and END, on INQUIRE TDQUEUE commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

ATIFACILITY(*cvda*) (intrapartition queues only)

Returns a CVDA value indicating whether the queue has a terminal (or session) associated with it. If it does, and CICS creates a task to process the

queue because its trigger level has been reached, the terminal is assigned as the principal facility of the task. See also the ATITERMID and ATITRANID options. CVDA values are as follows:

NOTAPPLIC

The queue is not intrapartition.

NOTERMINAL

No terminal is associated with the queue.

TERMINAL

A terminal is associated with the queue.

ATITERMID(*data-area*) (intrapartition queues only)

Returns the 4-character name of the terminal or session associated with the queue, if any. (See the ATIFACILITY option.) Otherwise, blanks are returned.

ATITRANID(*data-area*) (intrapartition queues only)

Returns the 4-character identifier of the transaction to be run when CICS initiates a task automatically to process the queue. This option applies only to intrapartition queues intended for ATI; for other types of queues, and for intrapartition queues where no transaction has been specified in the queue definition, the value returned is blanks.

ATIUSERID(*data-area*) (intrapartition queues only)

Returns the 8-byte user identifier associated with the queue. CICS assigns this value to a task that it creates to process the queue if no terminal is associated with the queue. If the queue is not intrapartition, or no transaction is defined for it with the ATITRANID option, blanks are returned.

If the security manager is not active, the value returned is that of the default user ID and not any value that has been included in the installed definition.

BLOCKFORMAT(*cvda*) (extrapartition queues only)

Returns a CVDA value indicating whether the data set associated with the queue is in blocked record format or not. It applies only to extrapartition queues. CVDA values are as follows:

BLOCKED

The records are blocked.

NOTAPPLIC

The data set is not open or the queue is not an extrapartition queue.

UNBLOCKED

The records are not blocked.

BLOCKSIZE(*data-area*)

Returns the length of the block in bytes, in the range 1 - 32767.

CHANGEAGENT(*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

SYSTEM

The resource definition was last changed by the CICS or CICSplex system.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see FORMATTIME in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

DATABUFFERS (*data-area*) (**extrapartition queues only**)

Returns the number of buffers, in the range 1 - 255, to be used by the transient data queue.

DDNAME (*data-area*) (**extrapartition queues only**)

Returns an 8-character identifier, padded with blanks if necessary, that can refer to a data set name used in the startup JCL.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

DISPOSITION (*cvda*) (**extrapartition queues only**)

Returns a CVDA value indicating the status of the associated data set. CVDA values are as follows:

MOD The system first assumes that the data set exists. For an existing data set, MOD causes the read/write mechanism to be positioned after the last record in the data set. The read/write mechanism is positioned after the last record each time the data set is opened for output.

If the system cannot find volume information for the data set on the DD statement, in the catalog, or passed with the data set from a previous step, the system assumes that the data set is being created in this job step. For a new data set, MOD causes the read/write mechanism to be positioned at the beginning of the data set.

NOTAPPLIC

The option does not apply because the queue is not open or is not an extrapartition queue.

OLD The data set existed before this job step.

SHARE

The data set existed before this job step and can be read by other concurrent jobs.

Note: You can use the abbreviation SHR when using CEDA to define this parameter.

DSNAME(*data-area*) (extrapartition queues only)

Returns a 1- to 44-character name that indicates an associated QSAM data set or DUMMY data set. This data area is blank if SYSOUTCLASS is used.

EMPTYSTATUS(*cvda*) (extrapartition queues only)

Returns a CVDA value indicating the state of the queue with regard to space. CICS detects a FULL condition only when a task attempts to add a record and there is no space, and detects EMPTY only when a task attempts to read and there are no records. Consequently, a value of NOTEMPTY is returned unless one of these conditions has been detected. EMPTYSTATUS applies only to extrapartition queues. CVDA values are as follows:

EMPTY

The queue is empty.

FULL The queue is full.

NOTAPPLIC

The option does not apply because the queue is not open or is not extrapartition.

NOTEMPTY

No operation against the queue has indicated that it is either empty or full.

ENABLESTATUS(*cvda*) (all except indirect queues)

Returns a CVDA value indicating whether the queue can be accessed by applications. For remote queues, this value reflects whether the local CICS will forward commands to access the queue to the remote system or reject them with a DISABLED exception condition; it does not necessarily reflect the state of the queue on the remote system. CVDA values are as follows:

DISABLE PENDING

The queue is currently being disabled.

DISABLED

The queue cannot be accessed by applications. For extrapartition queues, this value does not necessarily mean that the associated data set is closed.

DISABLING

The queue is currently being disabled.

ENABLED

The queue can be accessed by applications.

NOTAPPLIC

The queue is indirect.

ERROROPTION(*cvda*) (extrapartition queues only)

Returns a CVDA value indicating the action that CICS takes if an I/O error is encountered. CVDA values are as follows:

IGNORERR

The block that caused the error is accepted.

SKIP The block that caused the error is skipped.

INDIRECTNAME(*data-area*) (indirect queues only)

Returns the 4-character name of the queue to which this indirect queue points. This option applies only to queues defined as indirect; for other types of queues, blanks are returned.

INDOUBT(*cvda*) (intrapartition queues only)

Returns a CVDA value indicating the action that CICS is to take for an indoubt unit of work (UOW) if the definition for this queue specifies WAIT(YES). CVDA values are as follows:

QUEUE

The UOW is indoubt and waiting; any locks held by the UOW for this queue remain active until the final state of the UOW is known. Tasks are suspended rather than receiving the LOCKED response. When the final state of the UOW is known, any changes that it has made are committed or backed out. Until then, any further requests of the following types that need one of the active locks must wait:

- READQ if the indoubt UOW had issued READQ or DELETEDQ requests
- WRITEQ if the indoubt UOW had issued WRITEQ or DELETEDQ requests
- DELETEDQ if the indoubt UOW had issued READQ, WRITEQ, or DELETEDQ requests

REJECT

The UOW is indoubt and waiting, and any locks held by the UOW for this queue are retained until the final state of the UOW is known. When the final state is known, any changes it has made are committed or backed out. Until then, any further requests that need one of the retained locks are rejected, and a LOCKED condition is returned. REJECT causes LOCKED to be raised in exactly the same circumstances as those in which QUEUE causes a transaction to wait.

INDOUBTWAIT(*cvda*) (intrapartition queues only)

Returns a CVDA value indicating whether an indoubt unit of work (UOW), which has modified a recoverable queue, will wait for resynchronization with its coordinator to determine whether to commit or back out the changes. CVDA values are as follows:

NOWAIT

The UOW is not to wait, and any changes made to recoverable resources are to be backed out or committed, as specified by the ACTION attribute on the transaction resource definition.

WAIT The UOW is to wait and any action required while waiting is determined by the WAITACTION option.

This parameter overrides the WAIT option defined on the transaction definition of the UOW. See the *CICS Resource Definition Guide* for an explanation of the interactions of indoubt attributes on the TDQUEUE and TRANSACTION definitions.

INSTALLAGENT(*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

SYSTEM

The resource was installed by the CICS or CICSplex SM system.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

IOTYPE(*cvda*) (**extrapartition queues only**)

Returns a CVDA value indicating whether the queue was defined for INPUT, OUTPUT, or RDBACK. CVDA values are as follows:

INPUT

The queue is defined for input and is read forward.

NOTAPPLIC

The queue is not open or is not an extrapartition queue.

OUTPUT

The queue is defined for output.

RDBACK

The queue is defined for input and is read backward.

MEMBER(*data-area*) (**extrapartition queues only**)

Returns the 8-character member name if the queue is a member of a partitioned data set. If not, blanks are returned.

NUMITEMS(*data-area*) (**intrapartition queues only**)

Returns a fullword binary field giving the number of items in the queue. A value of -1 is returned if the queue is not intrapartition.

OPENSTATUS(*cvda*) (**extrapartition queues only**)

Returns a CVDA value indicating whether the queue is open, closed, or in an intermediate state. CVDA values are as follows:

CLOSED

The queue is closed.

CLOSING

The queue is closing.

NOTAPPLIC

The queue is not extrapartition.

OPEN The queue is open.**OPENING**

The queue is opening.

PRINTCONTROL(*cvda*) (**extrapartition queues only**)

Returns a CVDA value indicating the type of print control, if any, defined for the queue. Printer control characters appear in the first position of every record when used. However, CICS does not check this character when records are written to the queue, or remove the character when records are read from the

queue; use and enforcement of the printer control conventions are up to the applications using the queue. CVDA values are as follows:

ASACTL

ASA control characters are used.

MCHCTL

Machine control characters are used.

NOCTL

No print control characters are used.

NOTAPPLIC

The queue is not open or is not extrapartition.

RECORDFORMAT(*cvda*) (extrapartition queues only)

Returns a CVDA value indicating whether the queue has fixed- or variable-length records. CVDA values are as follows:

FIXED

The queue has fixed-length records.

NOTAPPLIC

The queue is not open or is not extrapartition.

VARIABLE

The queue has variable-length records.

RECORDLENGTH(*data-area*) (extrapartition queues only)

Returns a fullword binary field giving the record length in bytes for queues having fixed-length records, or the maximum record length for queues having variable-length records. The RECORDLENGTH option applies only to extrapartition queues; for others, -1 is returned.

RECOVSTATUS(*cvda*) (intrapartition queues only)

Returns a CVDA value indicating the type of recovery defined for the queue. Recovery is available only for intrapartition queues. CVDA values are as follows:

LOGICAL

The queue is logically recoverable.

NOTAPPLIC

The queue is not intrapartition.

NOTRECOVERABLE

The queue is not recoverable.

PHYSICAL

The queue is physically recoverable.

REMOTENAME(*data-area*) (remote queues only)

Returns the 4-character name of this queue in the remote CICS region in which the queue is defined (from the RMTNAME option in its definition). The REMOTENAME option applies only to queues defined as remote; for other queues the value returned is blanks.

REMOTESYSTEM(*data-area*) (remote queues only)

Returns the 4-character name of the CICS region in which the queue is defined (from the SYSIDNT value in its definition). The REMOTESYSTEM option applies only to queues defined as remote; for other queues the value returned is blanks.

REWIND(*cvda*) (extrapartition queues only)

Returns a CVDA value indicating the disposition of a tape data set. CVDA values are as follows:

LEAVE

The current tape is positioned to the logical end of the data set.

REREAD

The current tape is positioned to reprocess the data set.

SYSOUTCLASS(*data-area*)

Returns a single character indicating the class attribute of the associated SYSOUT data set or blank if DSNAME is used.

TDQUEUE(*data-value*)

Specifies the 4-character name of the transient data queue about which you are inquiring.

TRIGGERLEVEL(*data-area*) (intrapartition only)

Returns a fullword binary field giving the number of items that the queue must reach before automatic transaction initiation (ATI) occurs. When the queue reaches this depth, CICS calls a task to process it automatically. A value of zero means that the queue is not subject to ATI; a value of -1 is returned if the queue is not intrapartition.

TYPE(*cvda*)

Returns a CVDA value identifying the type of queue. CVDA values are as follows:

EXTRA

The queue is extrapartition.

INDIRECT

The queue is indirect.

INTRA

The queue is intrapartition.

REMOTE

The queue is remote.

Conditions**ENDCOND**

RESP2 values:

2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

1 Browse sequence error

NORMAL

RESP2 values:

0 No errors

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

QIDERR

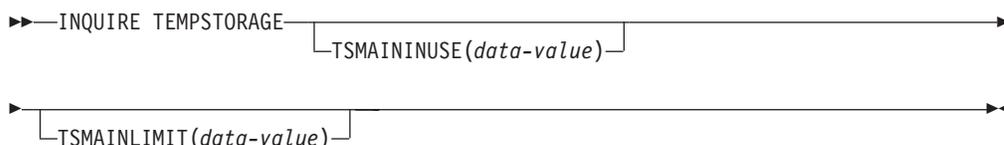
RESP2 values:

- 1 The named queue cannot be found.

INQUIRE TEMPSTORAGE

Retrieve information about the storage that is used by temporary storage queues in the CICS region.

INQUIRE TEMPSTORAGE



Conditions: NOTAUTH

This command is threadsafe.

Description

The INQUIRE TEMPSTORAGE command returns information about the CICS region storage that main temporary storage queues use, and the maximum amount that is available for their use.

Options

TSMMAININUSE(data-value)

Returns a doubleword binary value that shows the amount of storage, in bytes, that is currently used by main temporary storage queues.

TSMMAINLIMIT(data-value)

Returns a doubleword binary value that shows the current setting, in bytes, for the maximum amount of storage that CICS makes available for main temporary storage queues to use.

Conditions

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

INQUIRE TERMINAL

Retrieve information about a terminal or session.

INQUIRE TERMINAL or NETNAME



The following options apply to both the INQUIRE TERMINAL and the INQUIRE NETNAME command.

INQUIRE TERMINAL

ACCESSMETHOD—(—cvda—)
ACQSTATUS—(—cvda—)
AIDCOUNT—(—data-area—)
ALTPAGEHT—(—data-area—)
ALTPAGEWD—(—data-area—)
ALTPRINTER—(—data-area—)
ALTPRTCOPYST—(—cvda—)
ALTSCRNHT—(—data-area—)
ALTSCRNWD—(—data-area—)
ALTSUFFIX—(—data-area—)
APLKYBDST—(—cvda—)
APLTEXTST—(—cvda—)
ASCII—(—cvda—)
ATISTATUS—(—cvda—)
AUDALARMST—(—cvda—)
AUTOCONNECT—(—cvda—)
BACKTRANSST—(—cvda—)
COLORST—(—cvda—)
CONSOLE—(—data-area—)
COPYST—(—cvda—)
CORRELID—(—data-area—)
CREATESESS—(—cvda—)
DATASTREAM—(—cvda—)
DEFPAGEHT—(—data-area—)
DEFPAGEWD—(—data-area—)
DEFSCRNHT—(—data-area—)
DEFSCRNWD—(—data-area—)
DEVICE—(—cvda—)
DISCREQST—(—cvda—)
DUALCASEST—(—cvda—)
EXITTRACING—(—cvda—)
EXTENDEDSSST—(—cvda—)
FMHPARMST—(—cvda—)
FORMFEEDST—(—cvda—)
GCHARS—(—data-area—)
GCODES—(—data-area—)
HFORMST—(—cvda—)
HILIGHTST—(—cvda—)
KATAKANAST—(—cvda—)
LIGHTPENST—(—cvda—)
LINKSYSTEM—(—data-area—)
MAPNAME—(—data-area—)
MAPSETNAME—(—data-area—)
MODENAME—(—data-area—)
MSRCONTROLST—(—cvda—)
NATLANG—(—data-area—)

NATURE—(—cvda—)
NEXTTRANSID—(—data-area—)
NQNAME—(—data-area—)
OBFORMATST—(—cvda—)
OBOPERIDST—(—cvda—)
OPERID—(—data-area—)
OUTLINEST—(—cvda—)
PAGEHT—(—data-area—)
PAGESTATUS—(—cvda—)
PAGEWD—(—data-area—)
PARTITIONSST—(—cvda—)
PRINTADAPTST—(—cvda—)
PRINTER—(—data-area—)
PROGSYMBOLST—(—cvda—)
PRTCOPYST—(—cvda—)
QUERYST—(—cvda—)
RELREQST—(—cvda—)
REMTENAME—(—data-area—)
REMOTESYSNET—(—data-area—)
REMOTESYSTEM—(—data-area—)
SCRNHT—(—data-area—)
SCRNWD—(—data-area—)
SECURITY—(—cvda—)
SERVSTATUS—(—cvda—)
SESSIONTYPE—(—cvda—)
SIGNONSTATUS—(—cvda—)
SOSIST—(—cvda—)
TASKID—(—data-area—)
TCAMCONTROL—(—data-area—)
TERMMODEL—(—data-area—)
TERMPRIORITY—(—data-area—)
TERMSTATUS—(—cvda—)
TEXTKYBDST—(—cvda—)
TEXTPRINTST—(—cvda—)
TRACING—(—cvda—)
TRANSACTION—(—data-area—)
TTISTATUS—(—cvda—)
UCTRANST—(—cvda—)
USERAREA—(—ptr-ref—)
USERAREALEN—(—data-area—)
USERID—(—data-area—)
USERNAME—(—data-area—)
VALIDATIONST—(—cvda—)
VFORMST—(—cvda—)
ZCPTRACING—(—cvda—)

Conditions: END, ILLOGIC, NOTAUTH, TERMIDERR

For more information about the use of CVDAs, see CICS-value data areas (CVDA) in CICS System Programming Reference.

Description

The INQUIRE TERMINAL and INQUIRE NETNAME commands both return information about a particular terminal or session installed in a CICS region.

You can use these commands to inquire about any type of terminal resource, including these types:

- Physical terminals owned locally (by the region in which the INQUIRE is issued)
- Remote terminals (terminals defined locally as owned by another region)
- Surrogate terminals (partial definitions that represent terminals owned by another region, shipped to the local region the first time the definition is needed)
- Models (definitions used only to autoinstall other terminals)
- MVS consoles defined to CICS

Some of the options in this command return system status information, such as whether the terminal is acquired or not or whether it is in use by a task. Most options, however, reflect the definition of the terminal or session, modified, possibly, by subsequent SET TERMINAL commands or the information obtained from the hardware in a QUERY.

A terminal is specified by a TERMINAL resource definition and the TYPETERM definition to which it points. Characteristics shared by many terminals, such as screen size and 3270 features, are defined by TYPETERM, and those specific to one terminal, such as the name of the associated printer, are in the TERMINAL definition, which might have been autoinstalled. For a session, the CONNECTION defines shared properties and SESSIONS defines specifics.

In most cases, options of this type have the same name as the option (or a name that is similar name) to the option in the resource definition. Where this is not the case, the option descriptions that follow indicate the corresponding resource options.

INQUIRE NETNAME returns the same information as INQUIRE TERMINAL. With INQUIRE TERMINAL, you identify the object of your inquiry by providing its CICS terminal identifier in the TERMINAL option. NETNAME is optional. If you include it, CICS returns the network identifier in the data area you provide.

In an INQUIRE NETNAME command, the roles of TERMINAL and NETNAME are reversed. You identify the terminal about which you are inquiring by supplying its network identifier in NETNAME, and CICS returns the corresponding CICS terminal identifier in TERMINAL if you also include that option. TERMINAL must be before NETNAME (if present) in an INQUIRE TERMINAL command, and vice versa in an INQUIRE NETNAME command.

All of the other options apply to both commands and return the same information. Not all options apply to all types of terminals, however. In particular, when CICS ships a terminal definition from the owning region to a remote region, an inquiry issued in the owning region (where the definition is of a real terminal) produces more information than an inquiry issued in the remote region, where the definition is a *surrogate* for the one in the owning region.

Browsing

You can also browse through the definitions of all the terminals installed in your system by using the browse options (START, NEXT, and END) on INQUIRE TERMINAL or INQUIRE NETNAME commands. See Browsing resource definitions in CICS System Programming Reference for general information about browsing, including syntax, exception conditions, and examples.

Options

ACCESSMETHOD(*cvda*)

Returns a CVDA value indicating the access method defined for the terminal. Here are the CVDA values:

BGAM

The access method is BGAM.

BSAM

The access method is BSAM.

CONSOLE

The terminal is an operating system console, accessed through MVS console support facilities.

NOTAPPLIC

The terminal is an MRO session.

VTAM

The access method is z/OS Communications Server.

ACQSTATUS(*cvda*) (**z/OS Communications Server only**)

Returns the same value as the TERMSTATUS option and is retained only for compatibility purposes. Use TERMSTATUS in new applications.

AIDCOUNT(*data-area*)

Returns a fullword binary field giving the number of automatic initiate descriptors (AIDs) queued for the specified terminal. If there are no AIDs, then an AIDCOUNT value of 0 is returned.

ALTPAGEHT(*data-area*)

Returns a halfword binary field giving the height, in lines, of the alternate page size. See also the DEFPAGEHT and PAGEHT options.

ALTPAGEWD(*data-area*)

Returns a halfword binary field giving the width, in characters, of the alternate page size. See also the DEFPAGEWD and PAGEWD options.

ALTPRINTER(*data-area*)

Returns the 4-character name of the printer designated for print key requests and ISSUE PRINT commands from tasks at this terminal when the printer named in the PRINTER option of the TERMINAL definition is not available.

ALTPRTCOPYST(*cvda*)

Returns a CVDA value indicating whether CICS is to use the hardware copy feature to satisfy a print request on the printer named in the ALTPRINTER option. Here are the CVDA values:

ALTPRTCOPY

CICS is to use the hardware copy feature.

NOALTPRTCOPY

CICS is not to use the hardware copy feature.

NOTAPPLIC

The terminal is not a z/OS Communications Server terminal, or is a remote terminal, a surrogate terminal, or a model definition.

ALTSCRNHT(*data-area*)

Returns a halfword binary field giving the height, in lines, of the alternate screen size. See also the DEFSCRNHT and SCRNHT options.

ALTSCRNWD (*data-area*)

Returns a halfword binary field giving the width, in characters, of the alternate screen size. See also the DEFSCRNWD and SCRNWD options.

ALTSUFFIX (*data-area*)

Returns the 1-character suffix that BMS appends to map set names for maps written to this terminal when the screen is the alternate size and suffixing is in use.

If ALTSUFFIX was not specified in the definition of this terminal, the byte returned contains x'00'. Notice that the value x'00' is not described as null, because this field is a character field and, in that context, null refers to the blank character x'40', see for a discussion of null values.

APLKYBDST (*cvda*)

Returns a CVDA value indicating whether the terminal has the APL keyboard feature. Here are the CVDA values:

APLKYBD

The terminal has the APL keyboard feature.

NOAPLKYBD

The terminal does not have the APL keyboard feature.

APLTEXTST (*cvda*)

Returns a CVDA value indicating whether the terminal has the APL text feature. Here are the CVDA values:

APLTEXT

The terminal has the APL text feature.

NOAPLTEXT

The terminal does not have the APL text feature.

ASCII (*cvda*)

Returns a CVDA value indicating the type of ASCII code the terminal uses, if applicable. Here are the CVDA values:

ASCII7

The code is 7-bit ASCII.

ASCII8

The code is 8-bit ASCII.

NOTAPPLIC

The terminal does not use ASCII.

ATISTATUS (*cvda*)

Returns a CVDA value indicating whether CICS can initiate a task automatically (ATI) with this terminal as its principal facility.

ATI The terminal can be used in ATI.

NOATI

The terminal cannot be used in ATI.

AUDALARMST (*cvda*)

Returns a CVDA value indicating whether the terminal has the 3270 audible alarm feature. Here are the CVDA values:

AUDALARM

The terminal has the audible alarm feature.

NOAUDALARM

The terminal does not have the audible alarm feature.

AUTOCONNECT(*cvda*)

Returns a CVDA value indicating whether CICS attempts to establish (bind) a session with this terminal when communication with z/OS Communications Server is established. Here are the CVDA values:

ALLCONN

CICS binds the session. This value is returned when the AUTOCONNECT value is ALL in the associated TYPETERM definition (when you are inquiring about a terminal) or ALLCONN in the SESSIONS definition (when you are inquiring about a session).

AUTOCONN

CICS binds the session. This value is returned when the AUTOCONNECT value is YES in the associated TYPETERM definition (in an inquiry about a terminal) or AUTOCONN in the SESSIONS definition (in an inquiry about a session).

NONAUTOCONN

CICS does not bind a session.

NOTAPPLIC

The terminal is not a z/OS Communications Server terminal, or is a remote terminal, a surrogate, or a model.

BACKTRANSST(*cvda*)

Returns a CVDA value indicating whether the terminal has the 3270 background transparency feature. Background transparency allows you to control whether the display area behind a character is clear (transparent) or shaded. Here are the CVDA values:

BACKTRANS

The terminal has the background transparency feature.

NOBACKTRANS

The terminal does not have the background transparency feature.

COLORST(*cvda*)

Returns a CVDA value indicating whether the terminal has the 3270 extended color feature, which allows colors to be selected for individual fields or characters. Here are the CVDA values:

COLOR

The terminal has the extended color feature.

NOCOLOR

The terminal does not have the extended color feature.

CONSOLE(*data-area*)

Returns, for an MVS console only, a 12-byte string that identifies the console. If the device is not a console, CICS returns 12 blanks.

If the console is autoinstalled, or is defined explicitly with a console name, the name is returned in the first 8 bytes, and the last 4 bytes are blank.

COPYST(*cvda*)

Returns a CVDA value indicating whether the control unit through which the terminal is attached includes the copy feature. COPYST applies only to 3270 terminals. Here are the CVDA values:

COPY The control unit has the copy feature.

NOCOPY

The control unit does not have the copy feature.

CORRELID(*data-area*)

Returns an 8-character correlation-id that is set differently depending on the session:

- For LU6.1 sessions, it is set to the value of NETNAMEQ.
- For MRO sessions, it is set to the termid of the session at the other end of the MRO link to which this session is connected.
- For LU6.2 sessions, it is an 8-character token that is common to the two sessions that are connected.

Using CORRELID, you can relate the two parts of an MRO, LU6.1, or LU6.2 conversation, and so discover, for example, which program is running a particular function shipping mirror.

CREATESESS(*cvda*) (**z/OS Communications Server only**)

Returns a CVDA value indicating whether CICS attempts to acquire the terminal if it is required for an automatic task initiation (ATI) request. Only z/OS Communications Server physical terminals can be acquired by CICS; sessions are not eligible. Here are the CVDA values:

CREATE

The terminal can be acquired.

NOCREATE

The terminal cannot be acquired.

NOTAPPLIC

The terminal is not a z/OS Communications Server terminal or is a session (APPC, LUTYPE6.1, or MRO).

DATASTREAM(*cvda*)

Returns a CVDA value indicating the type of data stream used by the terminal. Here are the CVDA values:

DS3270

The terminal uses the 3270 data stream.

NOTAPPLIC

The terminal does not use either the 3270 or SCS data stream.

SCS

The terminal uses SNA character strings.

DEFPAGEHT(*data-area*)

Returns a halfword binary field giving the height, in lines, of the default page size. The corresponding option in the TYPETERM definition is PAGESIZE. See also the ALTPAGEHT and PAGEHT options.

DEFPAGEWD(*data-area*)

Returns a halfword binary field giving the width, in characters, of the default page size. The corresponding option in the TYPETERM definition is PAGESIZE. See also the ALTPAGEWD and PAGEWD options.

DEFSCRNHT(*data-area*)

Returns a halfword binary field giving the height, in lines, of the default screen size. See also the ALTSCRNHT and SCRNHT options.

DEFSCRNWD(*data-area*)

Returns a halfword binary field giving the width, in characters, of the default screen size. See also the ALTSCRNWD and SCRNWD options.

DEVICE(*cvda*)

Returns a CVDA value identifying the terminal or session type. CVDA values for this option are listed in CVDA values for the DEVICE option.

DISCREQST (*cvda*)

Returns a CVDA value indicating whether CICS is to honor a request to disconnect the terminal. Disconnect requests result from an **ISSUE DISCONNECT** command or a CESF (sign-off) task with the GOODNIGHT or LOGOFF option. The CVDA values are as follows:

DISCREQ

CICS will honor a request to disconnect this terminal, with a z/OS Communications Server CLSDST request to end the session if the terminal is a z/OS Communications Server terminal.

NODISCREQ

CICS will not honor a request to disconnect this terminal.

NOTAPPLIC

The option does not apply to this terminal.

DUALCASEST (*cvda*)

Returns a CVDA value indicating whether the terminal has a typewriter keyboard or an operator console keyboard. Here are the CVDA values:

DUALCASE

The terminal has a typewriter keyboard.

NODUALCASE

The terminal has an operator console keyboard or is not a 3270 display.

EXITTRACING (*cvda*) (**z/OS Communications Server only**)

Returns a CVDA value indicating whether this terminal is traced when CICS z/OS Communications Server exit tracing is active. See the TCEXITSTATUS option in the INQUIRE TRACEFLAG command. Here are the CVDA values:

EXITTRACE

The terminal is traced.

NOEXITTRACE

The terminal will not be traced.

NOTAPPLIC

The terminal is not a z/OS Communications Server terminal, or is a remote terminal, a surrogate terminal, or a model definition.

EXTENDEDSDSST (*cvda*)

Returns a CVDA value indicating whether the terminal supports the 3270 extended data stream. The terminal has this support if the TYPETERM definition specifies it either explicitly in the EXTENDEDSDS option or implicitly, by specifying features that use the extended data stream; see the BACKTRANST, COLORST, HIGHLIGHTST, MSRCONTROLST, OUTLINEST, PARTITIONSST, PROGSYMBOLST, SOSIST, and VALIDATIONST options of this command. Extended data stream support implies that the terminal accepts write-structured fields commands, including QUERY, and, conversely, support for QUERY; that is, a value of ALL or COLD for the QUERY option implies support for the extended data stream. Here are the CVDA values:

EXTENDEDSDS

The terminal supports the extended data stream.

NOEXTENDEDSDS

The terminal does not support the extended data stream.

FMHPARMST (*cvda*)

Returns a CVDA value indicating whether BMS accepts user-supplied values

for inclusion in a function management header (FMH) to be built by BMS. This support is available only on 3650 terminals. Here are the CVDA values:

FMHPARM

BMS allows user-supplied values.

NOFMHPARM

BMS does not allow user-supplied values.

FORMFEEDST (*cvda*)

Returns a CVDA value indicating whether the terminal has the forms feed feature. Here are the CVDA values:

FORMFEED

The terminal has the forms feed feature.

NOFORMFEED

The terminal does not have the forms feed feature.

GCHARS (*data-area*)

Returns a halfword binary field giving the graphic character set global identifier (GCSGID), which identifies the set of graphic characters that can be used as input or output at this terminal. The corresponding option in the TYPETERM definition is CGCSGID.

The GCHARS option applies only to graphic terminals; for others, 0 is returned.

GCODES (*data-area*)

Returns a halfword binary field giving the code page global identifier (CPGID), which identifies the EBCDIC code page that defines the code points for the characters that can be input or output at the terminal. The corresponding option in the TYPETERM definition is CGCSGID.

The GCODES option applies only to graphic terminals; for others 0 is returned.

HFORMST (*cvda*)

Returns a CVDA value indicating whether the terminal has the horizontal forms feature, which is required for use of horizontal tabbing when formatting documents for output. Here are the CVDA values:

HFORM

The terminal has the horizontal forms feature.

NOHFORM

The device does not have the horizontal forms feature.

HIGHLIGHTST (*cvda*)

Returns a CVDA value indicating whether the terminal has the 3270 extended highlighting facility, which enables fields or characters to be displayed in reverse-video, underlined, or blinking. Here are the CVDA values:

HILIGHT

The terminal has extended highlighting.

NOHILIGHT

The terminal does not have extended highlighting.

KATAKANAST (*cvda*)

Returns a CVDA value indicating whether the terminal is a Katakana terminal. Here are the CVDA values:

KATAKANA

The terminal is a Katakana terminal.

NOKATAKANA

The terminal is not a Katakana terminal.

LIGHTPENST(*cvda*)

Returns a CVDA value indicating whether the terminal has the 3270 selector pen feature. Here are the CVDA values:

LIGHTPEN

The terminal has the selector pen feature.

NOLIGHTPEN

The terminal does not have the selector pen feature.

LINKSYSTEM(*data-area*)

Returns the 4-character name of the connection that is the real link towards the TOR for a remote terminal entry, if it is available. It is not available if some connection definitions in the chain from the remote entry to the link system are missing.

MAPNAME(*data-area*)

Returns the 7-character name of the map that was most recently referenced in the MAP option of a SEND MAP command processed for this terminal. If this terminal is a surrogate and the terminal-owning system is a CICS Transaction Server for z/OS region, the map name might be the last map sent by the terminal-owning region or another AOR in which this terminal has been represented as a surrogate device. The map name returned might no longer be held in the device buffer, because an intervening BMS command such as SEND TEXT or SEND CONTROL (or a terminal control SEND command), or operator action, might have partially or completely removed the map display. If the terminal is not supported by BMS, for example, this terminal is a session or CICS has no record of any map being sent, the value returned is blanks.

MAPSETNAME(*data-area*)

Returns the 8-character name of the mapset that was most recently referenced in the MAPSET option of a SEND MAP command processed for this terminal. If the MAPSET option was not specified on the most recent request, BMS uses the map name as the mapset name. In both cases, the mapset name used can be suffixed by a terminal or alternate suffix. If this terminal is a surrogate, the mapset name might be the last mapset used by the terminal-owning region or another AOR in which this terminal has been represented as a surrogate device. If the terminal is not supported by BMS, for example, this terminal is a session or CICS has no record of any mapset being used, the value returned is blanks.

MODENAME(*data-area*) (APPC only)

Returns the 8-character name of the session group to which the session about which you are inquiring belongs (from the LOGMODE option of the SESSIONS definition). MODENAME applies only to APPC logical units; for other types, the value returned is blanks.

MSRCONTROLST(*cvda*)

Returns a CVDA value indicating whether the terminal has a magnetic slot reader. This feature is available only on 8775 and 3643 terminals. Here are the CVDA values:

MSRCONTROL

The terminal has a magnetic slot reader.

NOMSRCONTROL

The terminal does not have a magnetic slot reader.

NATLANG(*data-area*)

Returns a 1-character value giving the national language specified in the terminal definition. This value cannot be changed by any command and is not necessarily the same as the national language currently in use at the terminal. To determine current language, see the NATLANGINUSE option of the ASSIGN command. Possible values are listed in the *CICS RACF Security Guide*. A blank means that no value has been specified.

NATURE(*cvda*)

Returns a CVDA value identifying the nature of the terminal definition. Here are the CVDA values:

MODEL

A remote terminal definition, representing a terminal owned by another CICS region, which is not currently expanded into a surrogate.

REMSESSION

A remote session.

SESSION

A session.

SURROGATE

A remote terminal definition, representing a terminal owned by another CICS region, which is expanded into a surrogate.

TERMINAL

A physical terminal definition.

NETNAME(*data-area*)

Returns the 8-character network name of the terminal about which you are inquiring.

For a physical terminal, this name is the one by which this terminal is known to z/OS Communications Server. For ISC sessions, it is the name by which the session or session group, if there are parallel sessions, is known to z/OS Communications Server. For MRO sessions, it is the name used by the connected region to log on to the interregion communication program. For a remote terminal, it is the name by which the terminal is known to the z/OS Communications Server in the remote region. For a remote terminal routed from a pre-CICS Transaction Server for OS/390 region, NETNAME is blank.

If the netname is a z/OS Communications Server LU alias, it is different from the netname component of the NQNAME, which always contains the real netname.

The description above applies to the NETNAME option in an INQUIRE TERMINAL command. In an INQUIRE NETNAME command, the roles of NETNAME and TERMINAL are reversed. NETNAME specifies the name of the terminal or session about which you are inquiring to CICS, rather than returning information, and TERMINAL returns the corresponding terminal identifier if you use it. See the description of INQUIRE NETNAME in CICS System Programming Reference.

NEXTTRANSID(*data-area*)

Returns the 4-character identifier of the transaction to be run to process the next unsolicited input from this terminal. This value comes from the TRANSACTION value in the TERMINAL or SESSIONS definition, if one has been specified. If the value has not been specified, it was set by the previous task for which the terminal was principal facility (in the TRANSID option of its

final RETURN command) and is blank if that task did not specify a value or if an active task has the terminal as principal facility.

NQNAME (*data-area*)

Returns the 17-character network-qualified name for any terminal that received an NQNAME from z/OS Communications Server at logon time.

This name applies to local terminals only; remote terminals do not have a network-qualified name.

NQNAME, which is supported for problem determination purposes only, is returned for both autoinstalled and RDO-defined resources if it has been supplied by z/OS Communications Server. However, it is not catalogued for RDO-defined resources and is therefore not available on a restart until that resource logs on again.

If the resource is non-z/OS Communications Server or a remote terminal, NQNAME is blank. If the resource is a z/OS Communications Server resource but has not yet received an NQNAME, CICS returns the known netname.

OBFORMATST (*cvda*)

Returns a CVDA value indicating whether outboard formatting can be used for this terminal. Here are the CVDA values:

NOOBFORMAT

This terminal does not support outboard formatting.

OBFORMAT

This terminal supports outboard formatting.

OBOPERIDST (*cvda*)

Returns a CVDA value indicating whether CICS uses outboard operator identifiers to support the BMS routing facilities at this terminal. This option applies only to the 3790 and 3770 batch data interchange logical units. Here are the CVDA values:

NOOBOPERID

CICS does not use outboard operator identifiers.

OBOPERID

CICS uses outboard operator identifiers.

OPERID (*data-area*)

Returns the 3-character operator identification code of the user signed on at the terminal.

If the terminal is a surrogate terminal, this value might not be current; it represents the user signed on at the time the terminal definition was shipped from the owning CICS region to this one and, who might have signed off. The OPERID might also be different from that of the user currently signed on if it has been changed with the SET TERMINAL command.

OUTLINEST (*cvda*)

Returns a CVDA value indicating whether the terminal has the 3270 field outlining feature. Here are the CVDA values:

NOOUTLINE

The terminal does not support field outlining. This value is always returned for a model terminal.

OUTLINE

The terminal supports field outlining.

PAGEHT (*data-area*)

Returns a halfword binary field giving the height, in lines, of the current page size for the terminal. See the DEFPAGEHT and ALTPAGEHT options.

PAGESTATUS (*cvda*)

Returns a CVDA value indicating how pages of BMS messages with a disposition of PAGING will be delivered to the terminal. Here are the CVDA values:

AUTOPAGEABLE

Pages are written automatically in sequence.

PAGEABLE

Pages are written on request from the operator.

PAGEWD (*cvda*)

Returns a halfword binary field giving the width, in characters, of the current page size for the terminal. See also the DEFPAGEWD and ALTPAGEWD options.

PARTITIONSST (*cvda*)

Returns a CVDA value indicating whether the terminal supports partitions. Here are the CVDA values:

NOPARTITIONS

The terminal does not support partitions.

PARTITIONS

The terminal supports partitions.

PRINTADAPTST (*cvda*)

Returns a CVDA value indicating whether the terminal has the printer adapter feature. Here are the CVDA values:

NOPRINTADAPT

The terminal does not have a printer adapter.

PRINTADAPT

The terminal has a printer adapter.

PRINTER (*data-area*)

Returns the 4-character name of the preferred printer for print key requests and ISSUE PRINT commands from tasks at this terminal. This printer is used if available; if not, the printer named in the ALTPRINTER option is second choice.

PROGSYMBOLST (*cvda*)

Returns a CVDA value indicating whether the terminal supports the 3270 programmed symbol feature, which enables the terminal to use multiple character sets. Here are the CVDA values:

NOPROGSYMBOL

The terminal does not support programmable symbols.

PROGSYMBOL

The terminal supports programmable symbols.

PRTCOPYST (*cvda*)

Returns a CVDA value indicating whether CICS is to use the hardware copy feature to satisfy a print request on the printer named on the PRINTER option. Here are the CVDA values:

NOPRTCOPY

CICS is not to use the hardware copy feature.

NOTAPPLIC

The terminal is not a z/OS Communications Server terminal, or is a remote terminal, a surrogate terminal, or a model definition.

PRTCOPY

CICS is to use the hardware copy feature.

QUERYST(*cvda*)

Returns a CVDA value indicating whether and when CICS will use a QUERY structured field to determine the characteristics of the terminal. Here are the CVDA values:

ALLQUERY

The terminal is to be queried each time it is connected.

COLDQUERY

The terminal is to be queried only when it is first connected after an initial or cold start of CICS. The device characteristics are stored on the global catalog for use on subsequent warm and emergency starts.

NOQUERY

The terminal is not to be queried.

RELREQST(*cvda*) (**z/OS Communications Server only**)

Returns a CVDA value indicating whether CICS is to honor requests from z/OS Communications Server to release the terminal or session. Here are the CVDA values:

NORELREQ

CICS cannot release the logical unit, or the access method is not z/OS Communications Server.

RELREQ

CICS can release the logical unit.

NOTAPPLIC

The option does not apply to this terminal.

REMOTENAME(*data-area*)

Returns the 4-character name of this terminal in the remote CICS region in which it is defined. REMOTENAME applies only to terminals defined as remote; for others, the value returned is blanks.

REMOTESYSNET(*data-area*)

Returns the 8-character netname of the owning TOR, if the subject of this inquiry is a remote terminal. If it is blank, but the terminal is remote, the system named in the REMOTESYSTEM field has not been installed, and no value was specified for the REMOTESYSNET option when the terminal was defined.

REMOTESYSTEM(*data-area*)

Returns the first four characters of a connection, if the subject of the inquiry is a remote terminal. The named connection can be either a connection entry that links towards the TOR or an indirect connection that provides the netname of the TOR.

Otherwise this field is blank.

SCRNHT(*data-area*) (**or SCREENHEIGHT**)

Returns a halfword binary field giving the height, in lines, of the current screen size. See also the DEFSCRNHT and ALTSCRNHT options.

SCRNHT is a synonym for the SCREENHEIGHT option of earlier releases of CICS. For compatibility, CICS recognizes SCREENHEIGHT as equivalent.

SCRNWD(*data-area*) (or **SCREENWIDTH**)

Returns a halfword binary field giving the current width of the terminal screen, in characters. See the DEFSCRNWD and ALTSCRNWD options.

SCRNWD is a synonym for the SCREENWIDTH option of earlier releases of CICS. For compatibility, CICS recognizes SCREENWIDTH as equivalent.

SECURITY(*cvda*)

Returns a CVDA value indicating whether the terminal has preset security; that is, whether a USERID value has been specified in the TERMINAL or SESSIONS definition, so that it is permanently signed on. Here are the CVDA values:

NOPRESETSEC

The terminal does not have preset security.

PRESETSEC

The terminal has preset security.

SERVSTATUS(*cvda*)

Returns a CVDA value indicating whether the terminal is available for use, from the point of view of the local CICS system, which might be different from the system that owns the terminal. SERVSTATUS corresponds to the INSERVICE option in the TERMINAL definition. "Available" (INSERVICE) does not necessarily imply, for a z/OS Communications Server terminal, that the terminal is acquired. Here are the CVDA values:

GOINGOUT

The terminal is put in OUTSERVICE status as soon as some current work has completed and is not available to new tasks.

INSERVICE

The terminal is available.

OUTSERVICE

The terminal is not available.

SESSIONTYPE(*cvda*)

Returns a CVDA value identifying the type of the session about which you are inquiring. This option applies only to z/OS Communications Server sessions. Here are the CVDA values:

APPCPARALLEL

A parallel APPC session group.

APPCSINGLE

A single APPC session.

LU61 An LUTYPE6.1 session.

NOTAPPLIC

The terminal is not one of the above.

SIGNONSTATUS(*cvda*)

Returns a CVDA value identifying whether the terminal currently has a signed-on user. Here are the CVDA values:

SIGNEDOFF

The terminal does not have a signed-on user.

SIGNEDON

The terminal has a signed-on user.

SOSIST(*cvda*)

Returns a CVDA value indicating whether the terminal supports mixed EBCDIC and double-byte character set (DBCS) fields. Here are the CVDA values:

NOSOSI

The terminal does not support mixed fields.

SOSI The terminal supports mixed fields.

TASKID(*data-area*)

Returns a fullword binary field giving the number of the user task currently running at this terminal. Zero is returned if no task is using the terminal.

TCAMCONTROL(*data-area*)

Obsolete. TCAM terminals are not supported.

Here are the meanings:

00	Null
40	Intermediate part of message
F1	First part of message
F2	Last part of message
F3	Whole message
F4	Intermediate part of message, end of record
F5	First part of message, end of record
F6	Last part of message, end of record
F7	Whole message, end of record
FE	TCAM is not active
FF	Not applicable (non-TCAM terminal)

TERMINAL(*data-value*)

Specifies the 4-character name of the terminal or session about which you are inquiring in an **INQUIRE TERMINAL** command. In an **INQUIRE NETNAME** command, this option *returns* the terminal identifier that corresponds to the NETNAME value you specified. See the NETNAME option and the general information for this command.

TERMMODEL(*data-area*)

Returns a halfword binary field giving the terminal model number.

TERMPRIORITY(*data-area*)

Returns a fullword binary field giving the priority of the terminal relative to other terminals, in the range 0 - 255.

TERMSTATUS(*cvda*) (**z/OS Communications Server only**)

Returns a CVDA value indicating whether CICS is in session with the logical unit represented by this terminal. Here are the CVDA values:

ACQUIRED

CICS is in session with the logical unit.

ACQUIRING

The session is in the process of being acquired.

NOTAPPLIC

The terminal is not a z/OS Communications Server terminal.

RELEASED

CICS is not in session with the logical unit.

RELEASING

The session is in the process of being released.

TEXTKYBDST (*cvda*)

Returns a CVDA value indicating whether the terminal has the 3270 text-keyboard feature. Here are the CVDA values:

NOTEXTKYBD

The terminal does not have the text-keyboard feature.

TEXTKYBD

The terminal has the text-keyboard feature.

TEXTPRINTST (*cvda*)

Returns a CVDA value indicating whether the terminal has the 3288 text-print feature. Here are the CVDA values:

NOTEXTPRINT

The terminal does not have the text-print feature.

TEXTPRINT

The terminal has the text-print feature.

TRACING (*cvda*)

Returns a CVDA value indicating the type of tracing defined for this terminal. Here are the CVDA values:

SPECTRACE

Special tracing is specified.

STANTRACE

Standard tracing is specified.

For a task that has this terminal as its principal facility, this value is combined with the TRACING option value of the transaction the task is executing to determine whether tracing is standard, special, or suppressed.

If the transaction TRACING value is SUPPRESSED, no tracing occurs. Otherwise, tracing is special if either the terminal or the transaction specifies SPECTRACE, standard if both specify STANTRACE.

A TRACING value of STANTRACE is assigned when the terminal is defined. You can specify SPECTRACE only with a SET TERMINAL command or the CICS-supplied CETR transaction.

TRANSACTION (*data-area*)

Returns the 4-character identifier of the transaction being executed by the task for which this terminal is the principal facility. Blanks are returned if no task is currently running at the terminal.

TTISTATUS (*cvda*)

Returns a CVDA value indicating whether this terminal can initiate tasks by entering unsolicited input. Here are the CVDA values:

NOTTI

This terminal cannot initiate transactions.

TTI

This terminal can initiate transactions.

UCTRANST (*cvda*)

Returns a CVDA value indicating whether input from this terminal is translated to uppercase characters automatically, at the time of receipt. Translation can be suppressed, but only in a conversational task, when input is solicited with a RECEIVE or CONVERSE ASIS command. This value comes from the UCTRAN option of the TYPETERM definition associated with the terminal. The PROFILE definition also has a UCTRAN option, but that value is not relevant here. Here are the CVDA values:

NOUCTRAN

Input from this terminal is not translated to uppercase characters on receipt. It is translated before presentation to the task issuing a RECEIVE, however, if the PROFILE definition for the transaction being run specifies translation. See The effect of the UCTRAN parameters for information on how the UCTRAN options on the terminal and transaction profiles interact.

TRANIDONLY

This value is the same as NOUCTRAN, with one difference. If the input is unsolicited, and CICS needs to use the initial characters of the input to decide which transaction to run, that decision is made from a copy of the input that has been translated to uppercase characters. The data presented to the task is the same for both.

UCTRAN

The input is translated to uppercase characters on receipt. It is unaffected by the translation option in the PROFILE.

USERAREA(*ptr-ref*)

Returns the address of the terminal control table user area (TCTUA) for this terminal. If there is no TCTUA, the address returned is X'FF000000'.

USERAREALEN(*data-area*)

Returns a halfword binary field giving the length of the user area. Zero is returned if there is no user area.

USERID(*data-area*)

Returns the 8-character identifier of the user signed on at this terminal or session.

If no user is signed on, the default user ID, as specified in the DFLTUSER system initialization parameter, is returned.

USERNAME(*data-area*)

Returns the 20-character name of the user signed on at this terminal or session; that is, the name corresponding to the USERID option value. If the information, which is provided by the external security manager, is shorter than 20 bytes, CICS pads it to 20 with trailing blanks. Blanks are returned if no user is signed on.

VALIDATIONST(*cvda*)

Returns a CVDA value identifying whether the device has the extended validation feature, which allows you to request special processing of keyboard input, in addition to normal 3270 function. This feature is available only on 8775 and 3290 terminals. Here are the CVDA values:

NOVALIDATION

The terminal does not have the extended validation feature or is a model terminal.

VALIDATION

The terminal has the extended validation feature.

VFORMST(*cvda*)

Returns a CVDA value indicating whether the terminal has the vertical forms feature, which is required for use of vertical tabbing when formatting documents for output. Here are the CVDA values:

NOVFORM

The device does not have the vertical forms feature.

VFORM

The terminal has the vertical forms feature.

ZCPTRACING(*cvda*) (z/OS Communications Server only)

Returns a CVDA value indicating whether this terminal is traced when CICS tracing for z/OS Communications Server terminals is turned on. Here are the CVDA values:

NOTAPPLIC

The terminal is not a z/OS Communications Server terminal, or is a surrogate terminal or a model definition.

NOZCPTRACE

The terminal is not traced.

ZCPTRACE

The terminal is traced.

Conditions

END

RESP2 value:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 value:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

TERMIDERR

RESP2 value:

- 1 The named terminal cannot be found.

Remote terminals

In addition to links to local terminals or devices, some terminal definitions refer to remote terminals. A remote terminal is a terminal that is owned by another CICS system.

The owning system of a remote terminal is called the terminal-owning region (TOR). Note that different terminals can have different TORs. TOR here refers to the owning system for a particular terminal.

The way that the local system is connected to the TOR makes a difference to the information that the REMOTESYSTEM field returns.

If the TOR is *directly* connected to the local system, the REMOTESYSTEM field usually names the CONNECTION definition or first four characters of the IPCONN definition for the link.

If a remote terminal is on a system that is *not directly linked* to the TOR, the REMOTESYSTEM field can name one of two types of connection:

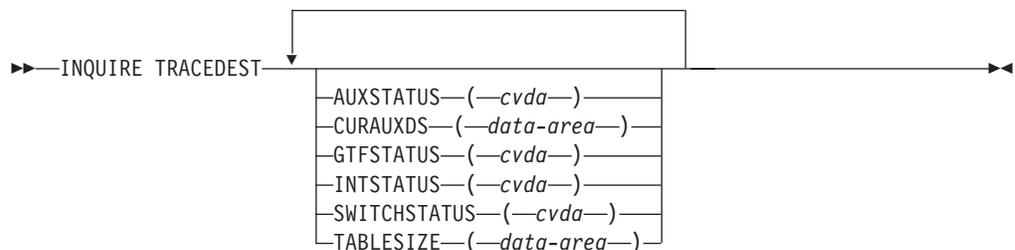
- A “real” connection that is the next link in the chain towards the TOR. In this case, the REMOTESYSNET option of the TERMINAL definition must specify the netname of the TOR.
- An indirect connection. In this case, the indirect connection NETNAME contains the netname of the TOR and its INDSYS option names another connection, which can also be indirect or “real”.

In both these cases, the LINKSYSTEM field of INQUIRE TERMINAL returns the “real” connection that is the next link towards the TOR. It is determined by looking at the logical chain of connections from the terminal in question to the “real” terminal entry. If the chain is broken, because a connection has not been installed yet, or has been discarded, LINKSYSTEM is not set. For an explanation of the relationship between REMOTESYSTEM, REMOTESYSNET, and LINKSYSTEM, see “Remote connections” on page 268.

INQUIRE TRACEDEST

Retrieve information about tracing.

INQUIRE TRACEDEST



Conditions: NOTAUTH

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The INQUIRE TRACEDEST command tells you where CICS trace entries are currently being written. There are three possible destinations, which can be used in any combination: the CICS internal trace table, the auxiliary trace data set, and the MVS Generalized Trace Facility (GTF). The number and types of trace entries are controlled by switch settings that you can determine with the INQUIRE TRACEFLAG and INQUIRE TRACETYPE commands.

Options

AUXSTATUS(*cvda*)

returns a CVDA value indicating whether auxiliary tracing is active; that is, whether trace entries are being written to an auxiliary trace data set. CVDA values are:

AUXPAUSE

Auxiliary tracing is not currently active, but was earlier in the current execution of CICS. It was suspended with a SET TRACEDEST AUXPAUSE command (or the CEMT equivalent). The current auxiliary

trace data set has been left open, and a subsequent SET TRACEDEST AUXSTART command will cause trace entries to be written immediately following those that were written before the AUXPAUSE request.

AUXSTART

Auxiliary tracing is active.

AUXSTOP

Auxiliary tracing is not active (the current trace data set, if any, is closed).

CURAUXDS (*data-area*)

returns the 1-character identifier of the current auxiliary trace data set, which can be 'A', 'B', or blank.

If your CICS system is initialized to allow auxiliary tracing, it will have either a single auxiliary trace data set, known as the 'A' data set, or two, 'A' and 'B'. The "current" or "active" one receives trace entries when auxiliary tracing is turned on, and the other, if there are two, is a standby, for use when the current one becomes full (see the SWITCHSTATUS option). If there is no auxiliary trace data set, the CURAUXDS value is blank.

GTFSTATUS (*cvda*)

returns a CVDA value indicating whether GTF tracing is active; that is, whether CICS is directing trace entries to the MVS Generalized Trace Facility (GTF). CVDA values are:

GTFSTART

GTF tracing is active.

GTFSTOP

GTF tracing is not active.

Note: In order to record trace entries on GTF, CICS must be initialized with GTF support (in the GTFTR system initialization option), GTF tracing must be started (with a SET TRACEDEST GTFSTART command or equivalent), and GTF trace must be started in MVS with the TRACE=USR option. If either of the first two conditions is not met, GTFSTATUS is GTFSTOP. However, GTFSTATUS can be GTFSTART without the third condition; in this case, no entries are written to GTF, but there is no other error indication.

INTSTATUS (*cvda*)

returns a CVDA value indicating whether internal tracing is active; that is, whether trace entries are being written in the internal trace table. CVDA values are:

INTSTART

Internal tracing is on.

INTSTOP

Internal tracing is off.

Note: Exception trace entries are always written to the internal trace table, regardless of the INTSTATUS value.

SWITCHSTATUS (*cvda*)

returns a CVDA value indicating the action that CICS is to take when the active auxiliary trace data set fills. If there are two data sets, CICS can switch them automatically when this occurs. Switching involves closing the current active data set, opening the standby, and reversing the designation of which is

active and standby. Without automatic switching, auxiliary tracing is stopped and cannot resume without a SET TRACEDEST command or the CEMT equivalent.

CVDA values are:

NOSWITCH

CICS takes no action.

SWITCHALL

CICS is to switch data sets every time the current one is full.

SWITCHNEXT

CICS is to switch data sets when the current one is full, but only once; thereafter NOSWITCH is in effect.

TABLESIZE(*data-area*)

returns a fullword binary field giving the size of the internal trace table in kilobytes.

Conditions

NOTAUTH

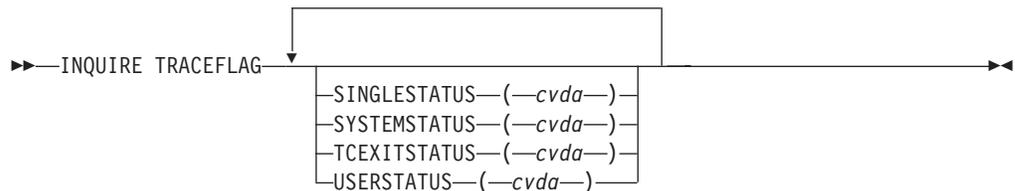
RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

INQUIRE TRACEFLAG

Retrieve information about trace flags.

INQUIRE TRACEFLAG



Conditions: NOTAUTH

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The INQUIRE TRACEFLAG command returns the current settings of the flags that control tracing in CICS generally, and for the task that issued the command specifically.

Tracing facilities and control are discussed in detail in the *CICS Problem Determination Guide*.

Options

SINGLESTATUS(*cvda*)

returns a CVDA value indicating whether tracing is turned on or is suppressed

for the task that issued this INQUIRE TRACEFLAG command. No non-exception trace entries are made for a task when this flag is off, regardless of the settings of the master trace flags (exception trace entries are *always* recorded).

The SINGLESTATUS value comes from the TRACE option in the definition of the TRANSACTION the task is executing, unless a different value has been specified, either for the transaction or for the terminal that is the principal facility, by means of the CICS-supplied CETR transaction. When a task is in progress, its SINGLESTATUS value can also be changed with a SET TRACEFLAG command.

CVDA values are:

SINGLEOFF

Tracing is suppressed.

SINGLEON

Tracing is allowed.

SYSTEMSTATUS(*cvda*)

returns a CVDA value indicating the status of the system master trace flag. This flag governs whether CICS makes or suppresses standard trace entries (it does not affect special or exception trace entries). It applies to all tasks and all system activity; however, for such trace entries to be recorded for any particular task, both the system master flag and the SINGLESTATUS flag for that task must be on.

CVDA values are:

SYSTEMOFF

Standard tracing is suppressed.

SYSTEMON

Standard tracing is active.

TCEXITSTATUS(*cvda*) (z/OS Communications Server only)

returns a CVDA value indicating which invocations of the CICS z/OS Communications Server exits are being traced.

Two types of exit activity can be traced: invocations associated with particular terminals that have been designated for z/OS Communications Server exit tracing (“terminal-specific” activity) and invocations not related to any particular terminal (“nonterminal-specific” activity).

CVDA values are:

NOTAPPLIC

z/OS Communications Server is not installed in the system.

TCEXITALL

All exit activity is being traced.

TCEXITNONE

No exit activity is being traced.

TCEXITSYSTEM

Nonterminal-specific activity is being traced, but terminal-specific activity is not.

USERSTATUS(*cvda*)

returns a CVDA value indicating the status of the user master trace flag. This flag governs whether non-exception user trace entries are recorded or suppressed (entries that specify the EXCEPTION option are never suppressed).

It applies to all tasks; however, for such entries to be recorded for any particular task, both the user master trace flag and the SINGLESTATUS flag for that task must be on. CVDA values are:

USEROFF

User tracing is suppressed.

USERON

User tracing is allowed.

Conditions

NOTAUTH

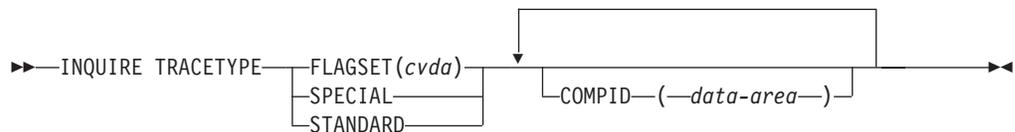
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

INQUIRE TRACETYPE

Retrieve information about CICS system tracing.

INQUIRE TRACETYPE



Conditions: INVREQ, NOTAUTH, NOTFND

Description

The **INQUIRE TRACETYPE** command tells you which levels of tracing are currently in effect for particular CICS system components.

Each CICS component has trace levels defined separately for standard CICS tracing and special tracing. (See the *CICS Problem Determination Guide* for definitions of these terms and for information about CICS tracing in general.) You can ask about either type of tracing in an **INQUIRE TRACETYPE** command, and for any number of components, but you can ask about only one type in a single command.

For each component that you specify, the trace levels are returned as a 32-bit string (in a 4-character data area). The bits are read from left to right; that is, the first bit is on if trace level 1 is in effect, bit 2 corresponds to trace level 2, and so on. A value of X'80000000' thus represents trace level 1. Bits for trace levels that are not defined are set to zero.

The SJ component uses higher trace levels than most other components. Trace levels 29 – 32 for the SJ component correspond to JVM Level 0 trace, JVM Level 1 trace, JVM Level 2 trace and JVM User trace respectively. The CICS-supplied transaction CETR includes screens that provide a better way to view the current settings and options for JVM trace.

Options

COMPID(*data-area*)

Returns the trace levels for the CICS component identified by COMPID in the format described above.

CICS components can be identified by a 2-character identifier or, in some cases, a descriptive keyword. For example, to determine the trace levels for the directory manager component of CICS, you can specify either:

```
INQUIRE TRACETYPE DD(data-area)
```

or

```
INQUIRE TRACETYPE DIRMGR(data-area)
```

The following list shows all the 2-character identifiers, and the keywords for those components that have them.

AP	APPLICATION	Application
BA	BUSAPPMGR	Business applications manager
BM*		Basic mapping support
BR*	BRIDGE	3270 Bridge
CP*	CPI	Common programming interface
DC*		Dump control
DD	DIRMGR	Directory manager
DH	DOCUMENT	Document handling
DM	DOMAINMGR	Domain manager
DP	DEBUGTOOL	Debugging Profiles domain
DS	DISPATCHER	Dispatch manager
DU	DUMP	Dump manager
EC*	EVENTCAPTURE	Event capture
EI*		EXEC interface
EJ	ENTJAVA	Enterprise Java domain
EM	EVENTMGR	Event manager
EP	EVENTPROC	Event processing domain
FC*		File control and DL/I
GC	GLOBALCATLG	CICS global catalog manager
IC*		Interval control
IE	IPECI	ECI over TCP/IP domain
II	IIOPI	IIOPI domain
IS*		Intersystem communication
KC*		Task control
KE	KERNEL	Kernel
LC	LOCALCATLG	CICS local catalog manager
LD	LOADER	Program load manager
LG	LOGGER	Log manager
LM	LOCKMGR	Lock manager
ME	MESSAGE	Message manager
ML		Markup language domain
MN	MONITOR	Monitoring manager
NQ	ENQUEUE	Enqueue domain
OT	OBJECTTRAN	Object Transaction Service (OTS) domain
PA	PARAMGR	Parameter manager
PC*		Program control
PG	PROGMGR	Program manager
PI	PIPEMGR	Pipeline manager domain
PT	PARTNER	Partner manager
RA*	RMIADAPTERS	Resource manager adapters

RI*	RMI	Resource manager interface (RMI)
RL	RESLIFEMGR	Resource life-cycle domain
RM	RECOVERY	Recovery manager
RS	REGIONSTAT	Region status
RX	RRS	Resource recovery services
RZ	REQUESTSTRM	Request streams domain
SC*		Storage control
SH	SCHEDULER	Scheduler services domain for BTS
SJ	SJVM	CICS JVM domain
SM	STORAGE	Storage manager
SO	SOCKETS	Sockets
ST	STATISTICS	Statistics manager
SZ*		Front-end programming interface
TC*		Terminal control
TD*		Transient data
TI	TIMER	Timer manager
TR	TRACE	Trace manager
TS	TEMPSTORAGE	Temporary storage
UE*		User exit interface
US	USER	User interface
WB	WEB	Web domain
WU	WEBRESTMGR	System Management RESTful API
W2	WEB2	Web 2.0 domain
XM	TRANMGR	Transaction manager
XS	SECURITY	Security manager

FLAGSET (*cvda*)

Indicates whether the standard or special flags for the specified component are to be returned. CVDA values are as follows:

SPECIAL

Indicates that CICS returns the trace levels for special tracing.

STANDARD

Indicates that CICS returns the trace levels for standard tracing.

Conditions

INVREQ

RESP2 values:

- 1 An incorrect value was specified for FLAGSET.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

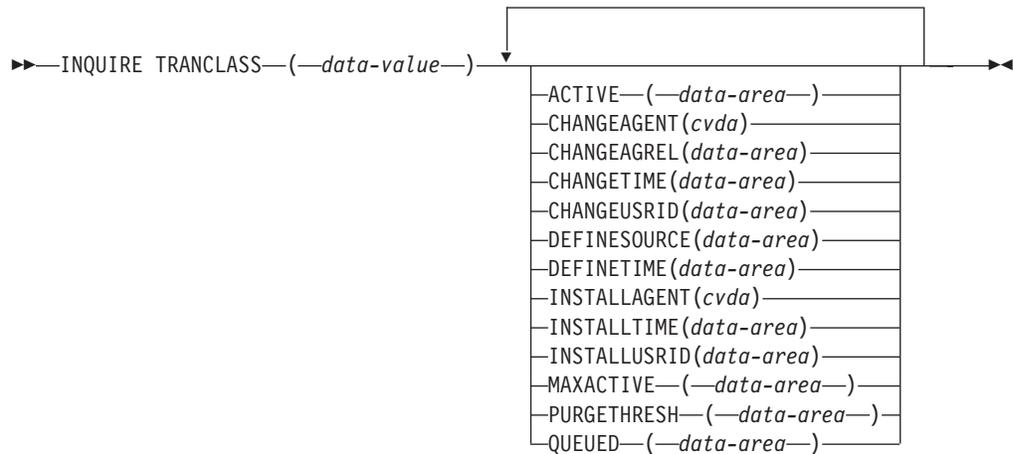
RESP2 values:

- 1 CICS was initialized without support for at least one of the components listed in the command; trace levels were returned for all other components.

INQUIRE TRANCLASS

Retrieve information about a transaction class.

INQUIRE TRANCLASS



Conditions: INVREQ, NOTAUTH, TCIDERR

Description

Use the INQUIRE TRANCLASS command to determine the limits defined for a transaction class and the current activity within the class.

Browsing

You can also browse through the definitions of all the transaction classes in your system by using the browse options, START, AT, NEXT, and END, on INQUIRE TRANCLASS commands. In browse mode, definitions are returned in alphabetical order, and you can specify a starting point with the AT option if you want. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

ACTIVE (data-area)

Returns a fullword binary field giving the current number of tasks in this class. This count does not include tasks that are queued waiting for initial dispatch.

CHANGEAGENT (cvda)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT** value. For more information, see **Summary of the resource signature field values in the Resource Definition Guide**.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID (*data-area*)

Returns the 8-character user ID that installed the resource.

MAXACTIVE (*data-area*)

Returns a fullword binary field giving the largest number of tasks in the transaction class that are allowed to run concurrently.

PURGETHRESH (*data-area*)

Returns a fullword binary field giving the maximum number of tasks in this

class that can be queued awaiting initial dispatch. See the QUEUED option. Tasks in this class that arrive while the queue is at its PURGETHRESH limit are purged.

QUEUED (*data-area*)

Returns a fullword binary field giving the number of tasks that are queued awaiting initial dispatch. Queuing occurs either because the number of active tasks is already at the maximum, or because the maximum for the system has been reached. See the MAXTASKS option in the INQUIRE SYSTEM command.

TRANCLASS (*data-value*)

Specifies the 8-character name of the transaction class about which you are inquiring. If the class is one of the numbered classes used in earlier releases of CICS, its name is DFHTCL*mm*, where *mm* is the two-digit class number.

Conditions

INVREQ

RESP2 values:

- 12 The TRANCLASS definition is in use.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

TCIDERR

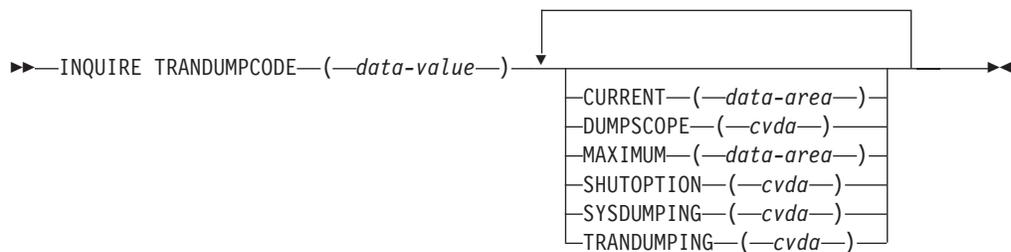
RESP2 values:

- 1 The transaction class cannot be found.

INQUIRE TRANDUMPCODE

Retrieve information about a transaction dump code.

INQUIRE TRANDUMPCODE



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The INQUIRE TRANDUMPCODE command allows you to look at some of the information in the transaction dump table entry for a particular transaction dump code.

The table entry tells CICS what actions to take when a transaction dump request with this code is received. Possible actions are: taking a transaction dump, taking a system dump (an MVS SDUMP), forwarding an SDUMP request to related MVS images, and shutting down CICS. The table entry also indicates how many times this set of actions is to be taken (the MAXIMUM option), and the number of requests received after the maximum are counted (the CURRENT option), but otherwise ignored.

CICS provides a transaction dump table with default actions for CICS transaction abend codes (those beginning with the letter *A*). These can be changed and others can be added with the SET TRANSDUMPCODE command or the CEMT transaction; such changes are preserved over executions of CICS, until an initial or cold start occurs.

CICS builds table entries, using default values, when it receives a dump request with a code for which it does not have an entry. You can also add your own entries with the SET TRANDUMPCODE command or a CEMT transaction.

Entries you add remain over executions of CICS until an initial or cold start occurs, but the entries CICS builds are considered temporary and are discarded at shutdown.

Consequently, if you enquire about a code that is not explicitly defined before it appears in a dump request, you get a “not found” response.

Browsing

You can also browse through all of the entries in the transaction dump table by using the browse options (START, NEXT, and END) on INQUIRE TRANDUMPCODE commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

Options

CURRENT (*data-area*)

returns a fullword binary field giving the number of dump requests with this dump code made since the count was last reset. (The count is reset automatically at CICS shutdown and can be reset explicitly with a SET SYSDUMPCODE RESET command or its CEMT equivalent.) The count includes requests that do not result in dumps, either because they are suppressed for this code or because the number for this code has reached its maximum.

DUMPSCOPE (*cvda*)

returns a CVDA value indicating whether a request for a dump with this dump code should cause an SDUMP (system dump) request to be sent to related MVS images.

A related image is one which contains a CICS region doing work on behalf of the task that caused the dump request - specifically, a region that has a task doing work under the same APPC token as the task causing the dump.

The sending of SDUMP requests occurs only when the table entry for this code specifies a system dump (that is, the SYSDUMPING value is SYSDUMP), and only in a sysplex environment executing under MVS/ESA 5.1 and the MVS workload manager.

CVDA values are:

LOCAL

SDUMP requests are not to be sent.

RELATED

SDUMP requests are to be sent.

Note: A setting of DUMPSCOPE(RELATED) results in a single dump being taken for each affected MVS image. This dump contains the output from all the affected CICS regions in the image. For more information, see the *CICS Problem Determination Guide*.

MAXIMUM(data-area)

returns a fullword binary field giving the maximum number of times CICS will take the set of actions indicated in the transaction dump table entry when a dump request with this code is received. A value of 999 means the default, 'no limit'.

SHUTOPTION(cvda)

returns a CVDA value indicating whether the CICS system is to be shut down after a request for a dump with this dump code. CVDA values are:

NOSHUTDOWN

The CICS system is not to shut down.

SHUTDOWN

The CICS system is to shut down.

SYSDUMPING(cvda)

returns a CVDA value indicating whether a system dump should be taken when a transaction dump request with this code is received. Even when the dump table entry specifies a system dump, however, one is taken only when the CURRENT value is no greater than the MAXIMUM, and system dumps are not suppressed system-wide (see the DUMPING option in the INQUIRE SYSTEM command). CVDA values are:

NOSYSDDUMP

A system dump is not to be taken.

SYSDUMP

A system dump is to be taken.

TRANDUMPCODE(data-value)

specifies the 4-character transaction dump code about which you are inquiring. A valid transaction dump code has no leading or imbedded blanks.

TRANDUMPING(cvda)

returns a CVDA value indicating whether a transaction dump should be taken when a transaction dump request with this code is received. Even when the dump table entry specifies a transaction dump, however, one is taken only when the CURRENT value is no greater than the MAXIMUM. CVDA values are:

NOTRANDUMP

The transaction dump is to be suppressed.

TRANDUMP

The transaction dump is to be taken.

Conditions**END**

RESP2 values:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

- 1 The dump code cannot be found.

INQUIRE TRANSACTION

Retrieve information about a TRANSACTION definition.

INQUIRE TRANSACTION

▶▶—INQUIRE TRANSACTION—(*—data-value—*)—————▶

←		→
←	BREXIT—(— <i>data-area</i> —) CHANGEAGENT (<i>cvda</i>) CHANGEAGREL (<i>data-area</i>) CHANGETIME (<i>data-area</i>) CHANGEUSRID (<i>data-area</i>) CMDSEC—(— <i>cvda</i> —) DEFINESOURCE (<i>data-area</i>) DEFINETIME (<i>data-area</i>) DTIMEOUT—(— <i>data-area</i> —) DUMPING—(— <i>cvda</i> —) FACILITYLIKE—(— <i>data-area</i> —) INDOUBT—(— <i>cvda</i> —) INDOUBTMINS—(— <i>data-area</i> —) INDOUBTWAIT—(— <i>cvda</i> —) INSTALLAGENT (<i>cvda</i>) INSTALLTIME (<i>data-area</i>) INSTALLUSRID (<i>data-area</i>) ISOLATEST—(— <i>cvda</i> —) PRIORITY—(— <i>data-area</i> —) OTSTIMEOUT—(— <i>data-area</i> —) PROFILE—(— <i>data-area</i> —) PROGRAM—(— <i>data-area</i> —) PURGEABILITY—(— <i>cvda</i> —) REMOTENAME—(— <i>data-area</i> —) REMOTESYSTEM—(— <i>data-area</i> —) RESSEC—(— <i>cvda</i> —) ROUTING—(— <i>cvda</i> —) ROUTESTATUS—(— <i>cvda</i> —) RTIMEOUT—(— <i>data-area</i> —) RUNAWAY—(— <i>data-area</i> —) RUNAWAYTYPE—(— <i>cvda</i> —) SCRNSIZE—(— <i>cvda</i> —) SHUTDOWN—(— <i>cvda</i> —) STATUS—(— <i>cvda</i> —) STORAGECLEAR—(— <i>cvda</i> —) TASKDATAKEY—(— <i>cvda</i> —) TASKDATALOC—(— <i>cvda</i> —) TCLASS—(— <i>data-area</i> —) TRANCLASS—(— <i>data-area</i> —) TRACING—(— <i>cvda</i> —) TRPROF—(— <i>data-area</i> —) TWASIZE—(— <i>data-area</i> —)	→

Conditions: END, ILLOGIC, INVREQ, NOTAUTH, TRANSIDERR

For more information about the use of CVDAs, see CICS-value data areas (CVDA) in CICS System Programming Reference.

Description

The INQUIRE TRANSACTION command retrieves information about a particular transaction installed in your CICS system.

Most of the values come from the TRANSACTION resource definition, but a few come from the profile definition to which it points. These values are noted in the

descriptions. See TRANSACTION definition attributes and PROFILE definition attributes for full details about the attributes of these two types of resources.

Many of the values produced by an INQUIRE TRANSACTION command are the same as those produced by the same-named options in an INQUIRE TASK command, when the task is running the transaction, because a task acquires most of its characteristics from the definition of the transaction. However, as noted in the description of that command, the values for a task also reflect the CICS system environment.

Furthermore, when a task is routed from one CICS to another, the transaction specified in the sending region might be different from the one that is run in the receiving region, so that an inquiry about its TRANSACTION value can produce different results in the sending and receiving regions. Indeed, in the case of dynamic routing, the transaction specified in the sending CICS (and shown as the TRANSACTION value in an INQUIRE TASK there) need not even be defined if the default processing for an undefined transaction code is dynamic routing.

Browsing

You can also browse through all of the TRANSACTION definitions in your system by using the browse options, START, AT, NEXT, and END, on INQUIRE TRANSACTION commands. In browse mode, the definitions are returned in alphabetic order, and you can specify a starting point with the AT option if you want. See Browsing resource definitions in CICS System Programming Reference for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

BREXIT(*data-area*)

Returns the 8-character name of the bridge exit defined by the BREXIT parameter of the named transaction resource definition.

If BREXIT is not defined, blanks are returned.

CHANGEAGENT(*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

SYSTEM

The resource definition was last changed by the CICS or CICSplex system.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see FORMATTIME in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

CMDSEC (*cvda*)

Returns a CVDA value indicating whether command security checking is performed for tasks running this transaction. CVDA values are as follows:

CMDSECNO

Command security checking is not performed.

CMDSECYES

Command security checking is performed.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The DEFINESOURCE value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

DTIMEOUT (*data-area*)

Returns a fullword binary field giving the deadlock time out value in seconds for a task running this transaction. CICS stops a task that waits for a locked resource longer than its deadlock timeout value.

DUMPING (*cvda*)

Returns a CVDA value indicating whether CICS takes a transaction dump if a task running this transaction stops abnormally. CVDA values are as follows:

NOTRANDUMP

No dump is taken.

TRANDUMP

A dump is taken.

This data value applies only to abend dumps and has no effect on DUMP TRANSACTION commands.

FACILITYLIKE (*data-area*)

Returns the 4-character name of the terminal defined by the FACILITYLIKE parameter in the PROFILE associated with the named transaction resource definition.

If FACILITYLIKE is not defined, blanks are returned.

INDOUBT (*cvda*)

Returns a CVDA value, based on the ACTION attribute of the TRANSACTION resource definition, indicating the action to be taken if the CICS region fails or loses connectivity with its coordinator while a unit of work is in the indoubt period.

The action depends on the values returned in the INDOUBTWAIT and INDOUBTMINS options; if INDOUBTWAIT returns WAIT, the action is not usually taken until the time returned in INDOUBTMINS expires. (For exceptions to this rule, see the INDOUBTWAIT option.)

CVDA values are as follows:

BACKOUT

All changes made to recoverable resources are to be backed out.

COMMIT

All changes made to recoverable resources are to be committed, and the unit of work marked as completed.

If a program uses the obsolete DTB option, which was replaced by INDOUBT, a CVDA value of NOTSUPPORTED is returned.

INDOUBTMINS (*data-area*)

Returns a fullword binary field giving the length of time, in minutes, after a failure during the indoubt period, before the transaction is to take the action returned in the INDOUBT field. The returned value is valid only if the unit of work is indoubt and INDOUBTWAIT returns WAIT.

INDOUBTWAIT (*cvda*)

Returns a CVDA value, based on the WAIT attribute of the TRANSACTION definition, indicating how CICS is to respond if a failure occurs while a unit of work (UOW) is in an indoubt state. CVDA values are as follows:

NOWAIT

The UOW is not to wait, pending recovery from the failure. CICS is to take immediately whatever action is specified on the ACTION attribute of the TRANSACTION definition.

WAIT The UOW is to wait, pending recovery from the failure, to determine whether recoverable resources are to be backed out or committed.

Even if INDOUBTWAIT returns WAIT, aspects of the UOW might force CICS to take an immediate decision; that is, to take immediately the action specified on the ACTION attribute of the transaction definition. This action can happen if, for example, the UOW contains one of these sessions:

- Subordinate LU6.1 sessions
- Subordinate MRO sessions to pre-CICS Transaction Server for z/OS systems.

For further information about the meaning of the ACTION and WAIT attributes of the TRANSACTION definition, see the *CICS Resource Definition Guide*.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

SYSTEM

The resource was installed by the CICS or CICSplex SM system.

INSTALLTIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID (*data-area*)

Returns the 8-character user ID that installed the resource.

ISOLATEST (*cvda*)

Returns a CVDA value indicating whether a task running this transaction will run isolated when isolation is active in the system.

Isolation limits the access, for both read and write, of user-key programs to task storage. A program running in user key on behalf of an isolated task can access the task storage of only that task, and this storage cannot be accessed by programs running in user key on behalf of other tasks. Isolation does not affect access by CICS-key programs and does not apply to storage with the SHARED attribute or any other non task storage.

Isolation must be turned on for the system as well as the transaction for a task to run isolated. See the TRANISOLATE option of the INQUIRE SYSTEM command. CVDA values are as follows:

ISOLATE

Tasks run isolated.

NOISOLATE

Tasks do not run isolated.

OTSTIMEOUT (*data-area*)

Returns a fullword data area containing the default period in seconds that an OTS transaction, created in an EJB environment under this CICS transaction, is allowed to run before sync point.

PRIORITY (*data-area*)

Returns a fullword binary field giving the priority of this transaction relative to other transactions in the CICS system, in the range 1 - 255.

PROFILE (*data-area*)

Returns the 8-character name of the profile definition for this transaction. The profile defines attributes that govern the interaction between a task running the transaction and the terminal or session which is its principal facility.

PROGRAM (*data-area*)

Returns the 8-character name of the first program called by a task running this transaction.

PURGEABILITY(*cvda*)

Returns a CVDA value indicating whether CICS is allowed to purge this task; that is, to end it abnormally. Purge requests come from SET TASK PURGE commands or CEMT equivalents, and CICS can generate them internally to reclaim resources to relieve a system stall condition. CVDA values are as follows:

NOTPURGEABLE

The task cannot be purged.

PURGEABLE

The task can be purged.

The PURGEABILITY value is set initially by the SPURGE option in the definition of the TRANSACTION this task is running.

REMOTENAME(*data-area*)

Returns the 8-character name by which this transaction is known in the remote system, if it is defined as a remote transaction. See TRANSACTION definition attributes for a fuller discussion of the length of REMOTENAME. Blanks are returned if the transaction is not remote.

REMOTESYSTEM(*data-area*)

Returns the first four characters of the remote system on which this transaction is defined, if it is defined as a remote transaction.

If the remote transaction is defined as DYNAMIC=YES, and the REMOTESYSTEM option is omitted, CICS returns the name of the local region.

Blanks are returned if the transaction is not remote.

RESSEC(*cvda*)

Returns a CVDA value identifying whether resource-level security checking is performed for a task running this transaction. CVDA values are as follows:

RESSECNO

Resource-level checking is not performed.

RESSECYES

Resource-level checking is performed.

ROUTING(*cvda*)

Returns a CVDA value indicating whether a task running this transaction is subject to dynamic routing. CVDA values are as follows:

DYNAMIC

The task can be routed dynamically.

STATIC

The task cannot be routed dynamically.

ROUTESTATUS

Returns a CVDA value indicating whether, if the transaction is the subject of an eligible START command, it is routed using the enhanced routing method. CVDA values are as follows:

NOTROUTABLE

If the transaction is the subject of a START command, it is routed using the "traditional" method.

ROUTABLE

If the transaction is the subject of an eligible START command, it is routed using the enhanced method.

For details of the enhanced and “traditional” methods of routing transactions invoked by EXEC CICS START commands, see the *CICS Intercommunication Guide*.

RTIMEOUT (*data-area*)

Returns a fullword binary field giving the read time out value for a task running this transaction, in seconds. CICS stops a task if it waits for input longer than its read time out value. This value is defined in the profile definition; see the PROFILE option.

RUNAWAY (*data-area*)

Returns a fullword binary field giving the “runaway task” time, in milliseconds, for tasks running this transaction. If a task keeps control of the processor for more than this interval, CICS assumes it is in a loop and stops it. If the value is zero, CICS does not monitor the task for a runaway condition.

RUNAWAYTYPE (*cvda*)

Returns a CVDA value indicating the source of the RUNAWAY option value for this transaction. CVDA values are as follows:

SYSTEM

The value is the current default for the system. See the ICVR option of the INQUIRE SYSTEM command.

USER The value was defined explicitly in the transaction definition.

SCRNSIZE (*cvda*)

Returns a CVDA value indicating whether a task running this transaction uses the alternate or the default screen size. This value is defined in the profile definition; see the PROFILE option. CVDA values are as follows:

ALTERNATE

The alternate screen size is to be used.

DEFAULT

The default screen size is to be used.

SHUTDOWN (*cvda*)

Returns a CVDA value indicating whether this transaction can be run during CICS shutdown by a task created to process unsolicited input. The transaction also can be run in this situation if it appears in the transaction list table (XLT) for shutdown. CVDA values are as follows:

SHUTDISABLED

The transaction cannot be run.

SHUTENABLED

The transaction can be run.

STATUS (*cvda*)

Returns a CVDA value indicating whether the transaction is available for use. CVDA values are as follows:

DISABLED

The transaction is not available for use.

ENABLED

The transaction is available for use.

STORAGECLEAR (*cvda*)

Returns a CVDA value indicating whether CICS clears storage that is released from a task running this transaction, to prevent other tasks accidentally viewing confidential data. CVDA values are as follows:

CLEAR

Storage is cleared.

NOCLEAR

Storage is not cleared.

TASKDATAKEY(*cvda*)

Returns a CVDA value indicating the key of the storage that CICS assigns to a task running this transaction. This storage includes task-lifetime storage; that is, the transaction work area (TWA) and the EXEC interface block (EIB), and the storage that CICS obtains on behalf of programs that run under the task.

CVDA values are as follows:

CICSDATAKEY

CICS-key storage is assigned.

USERDATAKEY

User-key storage is assigned.

TASKDATALOC(*cvda*)

Returns a CVDA value indicating whether task-lifetime storage for a task running this transaction is above or below the 16 MB line. Task-lifetime storage includes the EIB and TWA. CVDA values are as follows:

ANY Task-lifetime storage can be above or below the 16 MB line.

BELOW

Task-lifetime storage must be below the 16 MB line.

TCLASS(*data-area*)

Returns a fullword binary field giving the number of the transaction class to which the transaction belongs, if the task belongs to a numbered class. Zero is returned if the transaction does not belong to any class, and an INVREQ exception condition is raised if the transaction belongs to a class that does not correspond to a numbered class.

The TCLASS option is retained for compatibility with earlier releases of CICS, where transaction classes were numbered from 1 to 10. In this release, transaction classes have 8-character names, specified by the TRANCLASS value in the definition; see that option in this command.

A class is numbered only if its name is of the form DFHTCL*nn*, where *nn* is a number from 00 to 10, and this number is returned by the TCLASS option in this command. The TRANSACTION definition can contain a TCLASS value as well, to allow the same definition to be installed in a system running under an earlier release, but the TCLASS value is ignored in this release and does not need to correspond to the TRANCLASS value.

TRACING(*cvda*)

Returns a CVDA value indicating the type of tracing to be done for tasks running this transaction. CVDA values are as follows:

SPECTRACE

Tracing is to be special.

SPRSTRACE

Tracing is suppressed.

STANTRACE

Tracing is to be standard.

If this value is other than SPRSTRACE and the task has a principal facility, the tracing value for the task is determined from a combination of the TRACING

values for its terminal and the transaction it is running. In this case, tracing is special if either the terminal or the transaction specifies SPECTRACE, standard if both specify STANTRACE.

A TRACING value of STANTRACE is assigned when the transaction is defined. You can specify other values only with a SET TERMINAL command or the CICS-supplied CETR transaction.

TRANCLASS (*data-area*)

Returns the 8-character name of the transaction class to which this transaction belongs. If the transaction does not belong to any class, the value DFHTCL00 is returned.

TRANSACTION (*data-value*)

Specifies the 4-character name of the transaction definition about which you are inquiring.

TRPROF (*data-area*)

Returns the 8-character name of the profile definition used to define attributes associated with the session used for routing, if transaction routing occurs.

TWASIZE (*data-area*)

Returns a fullword binary field giving the size, in bytes, of the transaction work area (TWA) for this transaction.

Conditions

END

RESP2 values:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

INVREQ

RESP2 values:

- 3 The TCLASS option has been specified in this INQUIRE command, and the transaction belongs to a class that is not one of the numbered classes DFHTCL00 through DFHTCL10.

NORMAL

RESP2 values:

- 10 The profile definition associated with the transaction is not available.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

TRANSIDERR

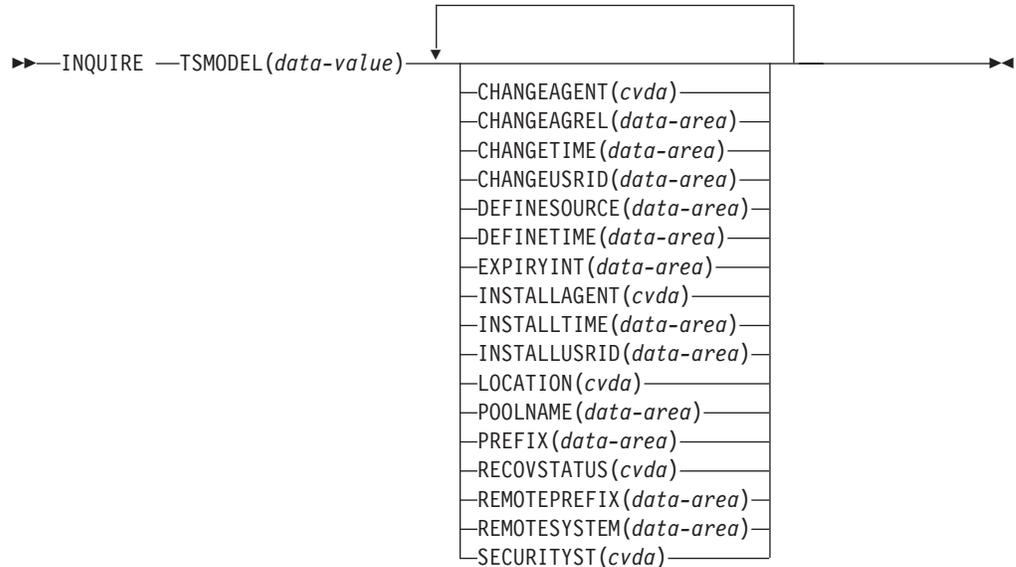
RESP2 values:

- 1 The transaction was not found.

INQUIRE TSMODEL

Retrieve information about a temporary storage model.

INQUIRE TSMODEL



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

For more information about the use of CVDAs, see CICS-value data areas (CVDA) in CICS System Programming Reference.

Description

The INQUIRE TSMODEL command returns information about a particular TS model.

Browsing

You can also browse the temporary storage models in your system by using the browse options (START, NEXT, and END) on INQUIRE TSMODEL commands.

See Browsing resource definitions in CICS System Programming Reference for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT** value. For more information, see **Summary of the resource signature field values in the Resource Definition Guide**.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

EXPIRYINT (*data-area*)

Returns a fullword binary field giving the expiry interval, in hours, for temporary storage queues matching this model. If a temporary storage queue is not referenced during its expiry interval, it becomes eligible to be deleted automatically by CICS. A value of zero means that no expiry interval applies to queues matching this model, so they are never eligible for automatic deletion. CICS does not apply an expiry interval to recoverable, remote, or shared temporary storage queues, or temporary storage queues created by CICS.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

LOCATION(*cvda*)

Returns a CVDA value indicating where queues matching the model are to be stored. CVDA values are as follows:

AUXILIARY

Queues matching this model are to be held on auxiliary storage.

MAIN Queues matching this model are to be held in main storage.

POOLNAME(*data-area*)

Returns an 8-character shared pool name.

PREFIX(*data-area*)

Returns a 16 byte character string or a 32 byte hexadecimal string with the value of the prefix for this model.

RECOVSTATUS(*cvda*)

Returns a CVDA value indicating the recovery status for this model. CVDA values are as follows:

RECOVERABLE

Queue names matching this model are recoverable.

NOTRECOVERABLE

Queue names matching this model are not recoverable.

REMOTEPREFIX(*data-area*)

Returns the 16 byte character string or 32 byte hex string to be used as the name prefix on the remote system.

REMOTESYSTEM(*data-area*)

Returns the 4-character name of the remote system on which the queues matching this model is defined.

SECURITYST(*cvda*)

Returns a CVDA value indicating the security status for this model. CVDA values are as follows:

SECURITY

Security checking is performed for queue names matching this model.

NOSECURITY

Security checking is not performed for queue names matching this model.

TSMODEL(*data-value*)

Specifies the 8-character name of a temporary storage model about which you are inquiring.

Conditions

END

RESP2 values:

2 There are no more resource definitions of this type.

Conditions

END

RESP2 values:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

NOTFND

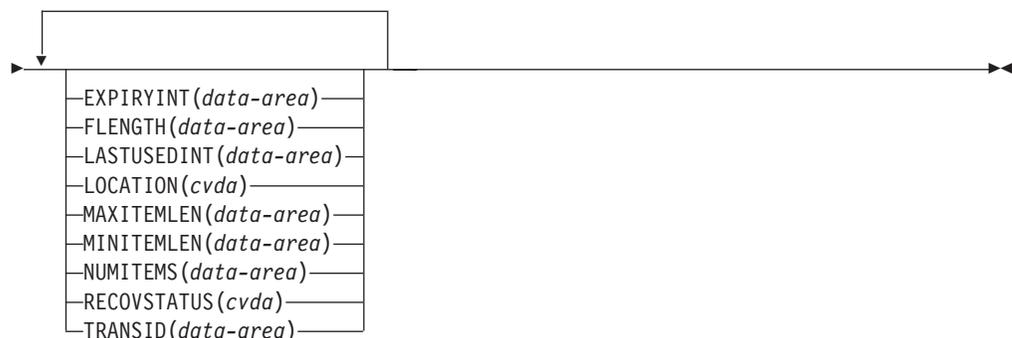
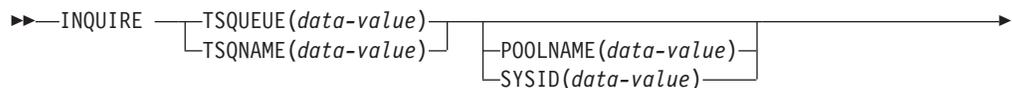
RESP2 values:

- 1 The TSPool does not exist.

INQUIRE TSQUEUE / TSQNAME

Retrieve information about a temporary storage queue. This topic applies also to the command INQUIRE TSQNAME. Use either command to inquire about names up to eight characters long; use INQUIRE TSQNAME to inquire about names up to 16 characters long.

INQUIRE TSQUEUE



Conditions: END, ILLOGIC, INVREQ, NOTAUTH, POOLERR, QIDERR, SYSIDERR

For more information about the use of CVDAs, see CICS-value data areas (CVDA) in CICS System Programming Reference.

Description

The INQUIRE TSQUEUE command returns information about a particular temporary storage queue.

Browsing

You can browse the temporary storage queues in your system by using the browse options (START, AT, NEXT, and END) on INQUIRE TSQUEUE commands. In browse mode, the definitions are returned in alphabetic order, and you can specify a starting point with the AT option if you want. For example, to see the queues with names that begin with ABC, start your browse with an AT value of ABC, padded on the right to eight characters with nulls (X'00').

To browse temporary storage queues that are in a shared temporary storage pool, you must specify the POOLNAME or the SYSID option on the browse START request only. You must specify an explicit SYSID for a queue that is defined by a TYPE=SHARED entry in a temporary storage table (TST) and that maps to a shared TS pool.

In a browse, CICS returns all queues, and you might see queues created by CICS for internal use as well as queues created by user applications. Queues with names that start with the following characters are CICS queues: **, \$\$, X'FA' through X'FF', CEBR, and DF.

See Browsing resource definitions in CICS System Programming Reference for general information about browsing, including syntax, exception conditions, and examples.

Options

EXPIRYINT(*data-area*)

Returns a fullword binary field that gives the expiry interval, in hours, that is defined for the temporary storage queue in its TSMODEL resource definition. If the temporary storage queue is not referenced during the expiry interval, it becomes eligible to be deleted automatically by CICS.

A value of zero means that no expiry interval applies to the temporary storage queue, so it is never eligible for automatic deletion. In addition, the following types of temporary storage queues are never deleted automatically by CICS, even if a nonzero expiry interval is set in the matching TSMODEL resource definition:

- Queues in auxiliary temporary storage that are defined as recoverable.
- Queues in a remote CICS region.
- Queues that CICS creates for its own use.
- Temporary storage queues in shared temporary storage pools.

FLENGTH(*data-area*)

Returns a fullword binary field that gives the total length in bytes of all the items in the temporary storage queue. For more information about queue lengths, see the MAXITEMLEN option.

For shared queues only: When the whole shared queue is stored in a single entry in the coupling facility, FLENGTH is the total size of all items including their control information. In this situation, the returned value for FLENGTH is less than 32K (32768).

When the shared queue has been stored as a separate list in the coupling facility, the total size is estimated as MAXITEMLEN multiplied by NUMITEMS.

LASTUSEDINT(*data-area*)

Returns a fullword binary field that gives the interval in seconds since the temporary storage queue was last referenced.

The value returned for large shared temporary storage queues is governed by the value of the LASTUSEDINTERVAL parameter specified for the associated TS queue manager. See Defining TS server regions.

LOCATION(*cvda*)

Returns a CVDA value indicating where the temporary storage queue resides. CVDA values are as follows:

AUXILIARY

The temporary storage queue is held in the CICS temporary storage VSAM data set (or in the coupling facility).

MAIN The temporary storage queue is held in main storage.

MAXITEMLEN(*data-area*)

Returns a halfword binary field that gives the length in bytes of the largest item in the temporary storage queue.

The length of a queue item is the sum of the length of the user data plus 8 bytes for header information, rounded up.

- For main temporary storage, the length is rounded up to the boundary of the MVS storage subpool used to store it.
- For auxiliary temporary storage, the length is rounded to the next highest multiple of either 64 or 128. The control interval size of the temporary storage data set determines which is chosen. See Control interval size for auxiliary temporary storage.
- For shared queues, the lengths returned in MINITEMLEN, MAXITEMLEN, and FLENGTH, reflect the data length stored in the coupling facility. The data length includes any item control information, which consists of a 2-byte length prefix for each item.

For all types of queue, the maximum value returned is capped at 32767 (X'7FFF').

MINITEMLEN(*data-area*)

Returns a halfword binary field that gives the length in bytes of the smallest item in the temporary storage queue.

The length of a queue item is the sum of the length of the user data plus 8 bytes for header information, rounded up.

- For main temporary storage, the length is rounded up to the boundary of the MVS storage subpool used to store it.
- For auxiliary temporary storage, the length is rounded to the next highest multiple of either 64 or 128. The control interval size of the temporary storage data set determines which is chosen. See Control interval size for auxiliary temporary storage.
- For shared queues, the lengths returned in MINITEMLEN, MAXITEMLEN, and FLENGTH, reflect the data length stored in the coupling facility. The data length includes any item control information, which consists of a 2-byte length prefix for each item.

For all types of queue, the maximum value returned is capped at 32767 (X'7FFF').

NUMITEMS(*data-area*)

Returns a halfword binary field that gives the number of items in the temporary storage queue.

POOLNAME(*data-value*) (TS data sharing only)

Specifies the name of a temporary storage pool. CICS ships the command to the temporary storage server that manages the pool.

For browse operations, specify POOLNAME on the browse START request only, not on the NEXT or END requests.

RECOVSTATUS(*cvda*)

Returns a CVDA value indicating the recovery status of the queue. CVDA values are as follows:

RECOVERABLE

The queue is recoverable.

NOTRECOVERABLE

The queue is not recoverable.

SYSID(*data-value*) (TS data sharing only)

Specifies the system name that corresponds to a temporary storage pool name. If CICS finds the specified system name in a TST TYPE=SHARED entry, it ships the command to the temporary storage server that manages the pool. If CICS does not find the system name in a TST, an INVREQ response is issued.

For browse operations, specify SYSID on the browse START request only, not on the NEXT or END requests.

TRANSID(*data-value*)

Specifies the identifier of the transaction that created the temporary storage queue.

TSQUEUE(*data-value*)

Specifies the 8-character name of the temporary storage queue about which you are inquiring.

TSQNAME(*data-value*)

Is an alternative to TSQUEUE and specifies the 16-character name of the temporary storage queue about which you are inquiring.

Conditions

END

RESP2 values:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

INVREQ

RESP2 values:

- 1 The specified SYSID does not exist in any TYPE=SHARED entry in the temporary storage table.

- 2 When INQUIRE TSQUEUE NEXT is specified, the next queue to be browsed has a queue name of more than eight significant characters. The queue name is truncated, and some significant characters are lost.
- 4 This temporary storage queue name cannot be deleted because it was written by CICS using the PUTQ macro.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

POOLERR

RESP2 values:

- 3 The POOLNAME does not exist.

QIDERR

RESP2 values:

- 1 The temporary storage queue cannot be found.

SYSIDERR

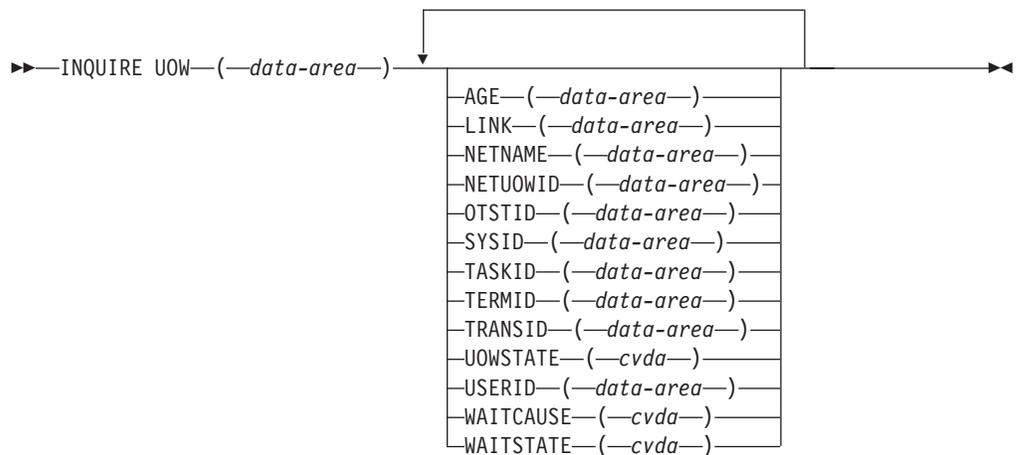
RESP2 values:

- 3 The SYSID does not map to a shared pool.
- 4 Server error.
- 5 I/O error on coupling facility.

INQUIRE UOW

Retrieve information about a unit of work (UOW).

INQUIRE UOW



Conditions: END, ILLOGIC, NOTAUTH, UOWNOTFOUND

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The INQUIRE UOW command retrieves information about a unit of work, or about all UOWs in a specified state. It returns the state of the UOW (for example, INDOUBT) and whether it is active, waiting, or shunted. In some cases, it returns the name of the resource that caused the UOW to be shunted, plus the transaction, user, and terminal that started it.

Important: In an intercommunication environment, a unit of work can include actions that are to be taken by two or more connected systems. Such a unit of work is known as a *distributed* unit of work, because the resources to be updated are distributed across more than one system. A distributed unit of work is made up of two or more *local* units of work, each of which represents the work to be done on one of the participating systems.

Note that INQUIRE UOW always returns information about *local* UOWs—that is, for a distributed UOW it returns information only about the work required on the system on which the command is issued. You can assemble information about a distributed UOW by matching the network-wide UOW identifier returned in the NETUOWID field against the network-wide identifiers of local UOWs on other systems.

For further information about local and distributed UOWs, see the *CICS Intercommunication Guide*.

Browsing

You can also browse through all of the UOWs currently in your system by using the browse options (START, NEXT, and END) on INQUIRE UOW commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

For example, if you suspect a problem with either a recoverable data set or a connection, you can use INQUIRE UOW to return information about UOWs that have been shunted because of a connection or data set failure.

Restriction: Do not issue SYNCPOINT commands during an INQUIRE UOW browse. The following sequence of commands causes an infinite loop:

```
EXEC CICS INQUIRE UOW START
  EXEC CICS INQUIRE UOW(data-area) NEXT
  SYNCPOINT
EXEC CICS INQUIRE UOW(data-area) NEXT
  SYNCPOINT
  ⋮
EXEC CICS INQUIRE UOW END
```

This is because every time the SYNCPOINT command is executed, a new UOW is created. The new UOW is returned in the next INQUIRE UOW, which is followed by a SYNCPOINT, and so on.

Options

AGE(*data-area*)

returns a fullword binary value giving the number of seconds since the UOW entered its current WAITSTATE.

LINK(*data-area*)

returns an 8-character value that, for a WAITCAUSE value of CONNECTION, is the netname of the remote system that caused the UOW to wait or be shunted. For other WAITCAUSE values, LINK returns blanks.

NETNAME(*data-area*)

returns the 8-character network name of the terminal from which the UOW was started. If the UOW was started from an ISC or MRO session, NETNAME returns the network name of the remote region. If the UOW was not started from a terminal, nor from an ISC or MRO session, NETNAME returns blanks. For OTS transactions, NETNAME returns blanks.

NETUOWID(*data-area*)

returns the LU6.2 name for the UOW within this network—that is, the network-wide identifier of the UOW. This is a 27-character data-area.

You can assemble information about a distributed UOW by matching the network-wide UOW identifier against the network-wide identifiers of local UOWs on other systems.

OTSTID(*data-area*)

returns the first 128 bytes of the transaction identifier (TID) of the OTS transaction which the UOW is a part. If the OTS name has fewer than 128 bytes, it is padded on the right with binary zeros.

SYSID(*data-area*)

returns a 4-character value that, for a WAITCAUSE value of CONNECTION, is the sysid of the connection that caused the UOW to wait or be shunted. If the connection has been discarded, and for other WAITCAUSE values, SYSID returns blanks.

TASKID(*data-area*)

returns a 4-byte packed-decimal value giving the task number originally associated with this UOW. If the UOW is shunted, the task terminates. In this case, the number may have been reused by another task.

TERMID(*data-area*)

returns the 4-character ID of the terminal or session from which this UOW was started. This is the principal facility for the task. If the transaction is the mirror transaction, CSML, it is the session. For UOWs that are part of an OTS transaction, TERMID is the session used by the request that attached the task.

TRANSID(*data-area*)

returns the 4-character ID of the transaction that started this UOW.

UOW(*data-area*)

specifies the 16-byte local identifier of the UOW about which you are inquiring, the last eight bytes of which are always null (X'00').

UOWSTATE(*cvda*)

returns a CVDA value indicating the state of the UOW. CVDA values are:

BACKOUT

This UOW is being backed out, or has failed to back out one or more of the recoverable resources involved in the UOW.

COMMIT

A decision to commit the UOW has been made, but the UOW is waiting or has been shunted. This may be because the decision has not yet been communicated to all participants in the syncpoint, or because a failure has occurred during commit processing.

FORCE

An attempt is being made to force the UOW to back out or commit, as specified on the ACTION option of the TRANSACTION resource definition.

HEURBACKOUT

The UOW has been forcibly backed out. A forced decision is taken when a UOW is unable to wait for indoubt resolution—for example, the transaction may have been defined as WAIT(NO), or backed out with a CEMT SET UOW command.

HEURCOMMIT

The UOW has been forcibly committed.

INDOUBT

This UOW is in the indoubt state.

INFLIGHT

The UOW is running normally.

USERID(*data-area*)

returns the 8-character user ID for which this transaction was running.

WAITCAUSE(*cvda*)

returns a CVDA value identifying the type of resource that caused the UOW to wait or be shunted.

Note: In the case of a wait, it is the UOW that is waiting, not the task.

Because each resource needs fields of the right type, WAITCAUSE also indicates which fields contain the RESOURCE NAME and QUALIFIER. CVDA values are:

CONNECTION

This UOW is waiting or has been shunted because of the failure of a session to the coordinator of the UOW during the indoubt period. NETNAME and SYSID contain the netname and system name of the failed link.

DATASET

This UOW is waiting or has been shunted because of a failure associated with one or more data sets. Use the INQUIRE UOWDSNFAIL command to identify the data sets involved and the reasons why they have caused the UOW to fail.

RLSSERVER

This UOW is waiting or has been shunted because of the failure of an RLS server.

WAITRRMS

This UOW is waiting or has been shunted because communication has been lost with RRS/MVS.

WAITCOMMIT

This UOW is waiting or has been shunted because a failure occurred during commit processing.

WAITFORGET

This UOW is waiting for FORGET from participants in the sync point. Use the INQUIRE UOWLINK command to obtain the netnames and sysids of the participants.

WAITRMI

This UOW is waiting for FORGET from the RMI. Use the INQUIRE UOWLINK command to obtain the entry name and qualifier of the task-related user exit.

WAITSTATE (*cvda*)

returns a CVDA value indicating whether the UOW is currently running or waiting. CVDA values are:

ACTIVE

The UOW is running normally.

SHUNTED

Sync point processing of the UOW has been deferred. A reason for this is returned in WAITCAUSE. SHUNTED further indicates that the task, terminal and program storage have been released, and locks have been retained.

WAITING

Sync point processing has completed on this system, but not on all systems involved in the distributed UOW. WAITCAUSE returns either WAITFORGET or WAITRMI, and UOWSTATE returns either BACKOUT or COMMIT to indicate how the UOW was resolved on this system.

Conditions**END**

RESP2 values:

- 2 All authorized resource definitions have been retrieved. All data areas specified on this command are left unchanged.

ILLOGIC

RESP2 values:

- 1 A browse of this resource type is already in progress, or an INQUIRE UOW START command has not been issued.

NOTAUTH

RESP2 values:

- 100 The use of this command is not authorized.

UOWNOTFOUND

RESP2 values:

- 1 The named UOW cannot be found.

INQUIRE UOWDSNFAIL

Retrieve information about units of work (UOWs) that have updated CICS file control-managed data sets.

INQUIRE UOWDSNFAIL

►► INQUIRE UOWDSNFAIL — CAUSE — (—cvda—) — DSNAME — (—data-area—) —►

DFH0BAT8, for a working illustration of the use of this command. These are supplied in the CICSTS42.CICS.SDFHSAMP library.

The INQUIRE UOWDSNFAIL function is in effect a two dimensional, or nested, browse: the first (outer) browse loops through all the UOWs, and within each UOW, the second (inner) browse loops through all the failed data sets associated with that UOW. Note that, in common with all browse functions, CICS does not lock resources during a browse operation. For each failed UOW, CICS obtains a snapshot of all the data sets that are failed for the UOW, and returns one UOW/data set pair for each NEXT operation. It is theoretically possible that the status of some data sets associated with an INQUIRE UOWDSNFAIL NEXT command could have changed by the time the information is returned to your program.

Options

CAUSE(*cvda*)

Returns a CVDA value that indicates which failed component has caused the UOW to have retained locks for this data set. CVDA values are as follows:

CACHE

A VSAM RLS cache structure, or a connection to it, has failed.

CONNECTION

An intersystem connection error has caused the UOW to fail while indoubt. The name of the system to which connectivity was lost is returned on the SYSID parameter and its netname is returned on the NETNAME parameter. CICS returns additional information in the REASON parameter about the connection failure.

DATASET

The backout of a UOW has failed for this data set. The reason for the data set failure is returned in the REASON parameter.

RLSSERVER

The SMSVSAM server has failed. The reason for the data set failure is returned in the REASON parameter.

UNDEFINED

The UOW is probably being retried. This can occur following a SET DSN RETRY command, or automatically when the failed resource returns. It can also occur following an emergency restart.

DSNAME(*data-area*)

Returns, as a 44-character value, the data set name of a data set that has experienced a backout failure in this UOW.

NETNAME(*data-area*)

Returns the 8-character netname (when the CVDA on the CAUSE parameter is CONNECTION) of the remote system to which connectivity has been lost.

REASON(*cvda*)

Returns a CVDA value (when the CVDA returned on the CAUSE parameter is RLSSERVER, CONNECTION, or DATASET) that indicates the specific reason for the error against this data set. CVDA values are as follows:

BACKUPNONBWO

Backout of the updates made to the data set by the UOW failed because a non-BWO backup of the data set was in progress while the UOW was being backed out. When the backup completes, CICS automatically retries the UOW.

COMMITFAIL

An error occurred at some point when RLS locks were in the process of being released. This is an error that can normally be resolved by recycling the SMSVSAM server (which should happen automatically). The locks were acquired as a result of recoverable requests having been issued against the data set.

DATASETFULL

No space is available on the direct access device for adding records to a data set. You need to reallocate the data set with more space. You can then retry the backout using SET DSNNAME RETRY..

DEADLOCK (non-RLS data sets only)

A deadlock was detected during backout. This is a transient condition that will probably go away if the backout is retried.

DELEXITERROR

Backout of a write to an ESDS failed because a logical delete global user exit program was not enabled, or a logical delete global user exit program decided not to execute the logical delete.

FAILEDBKOUT

This occurs as a result of a severe error being identified during backout, and is possibly an error in either CICS or VSAM. The problem may go away if the backout is retried. Note that CICS performs some first-failure data capture (FFDC) at the point where the error is first detected.

INDEXRECFULL

A larger alternate index record size needs to be defined for the data set..

This error can also occur when a unique alternate index key, for a non-RLS data set, has been reused and CICS is now backing out the request which had removed that key value.

INDOUBT

The unit of work had issued recoverable requests against the data set, and has now failed indoubt. The connection to the coordinating system needs to be reestablished.

IOERROR

A hard I/O error occurred during backout. To correct this error, restore a full backup copy of the data set and perform forward recovery. If you use CICS VSAM Recovery as your forward recovery utility, the backout is automatically retried for an RLS data set. For a non-RLS data set, use the SET DSNNAME (...) RETRY command to drive the backout retry.

LCKSTRUCFULL

An attempt to acquire a lock during backout of an update to this data set failed because the RLS lock structure was full. You must allocate a larger lock structure in an available coupling facility and rebuild the existing lock structure into it, then use the SET DSNNAME (...) RETRY command to drive the backout retry.

NOTAPPLIC

The CVDA for CAUSE is not CONNECTION, RLSSERVER, or DATASET.

OPENERERROR

Error on opening the file for backout. A console message notifies you of the reason for the open error. One likely reason could be that the data set was quiesced.

RLSGONE

An error occurred when backing out the UOW, because the SMSVSAM RLS server was inactive. This may also be the reason why the UOW went into backout originally. This is an error that can be resolved by recycling the server (which should happen automatically). Generally, when the server recovers, the UOWs are retried automatically. In very exceptional circumstances, it may be necessary to issue a SET DSNAME(...) RETRY command to retry UOWs that were not retried when the server returned.

RRCOMMITFAIL

An error occurred while RLS locks for the unit of work were being released. For this data set, the locks being released were all repeatable read locks, so if the failure was due to the RLS server being unavailable, the locks will have been released. If the failure was due to some other error from the SMSVSAM server, the locks may still be held.

RRINDOUBT

The unit of work had issued repeatable read requests against the data set, and has now failed with an indoubt condition. The locks will have been released, so this failure does not prevent you from running a batch job against the data set. However, if you want to open the data set in non-RLS mode from CICS, you need to resolve the indoubt failure before you can define the file as having RLSACCESS(NO). If the unit of work has updated any other data sets, or any other resources, you should try to resolve the indoubt failure correctly. If the unit of work has only performed repeatable reads against VSAM data sets and has made no updates to other resources, it is safe to force the unit of work using the SET DSNAME or SET UOW commands.

Each REASON (except for NOTAPPLIC) corresponds with only one CAUSE value. The mappings are as follows:

Cause	Reason
CACHE	NOTAPPLIC
CONNECTION	INDOUBT
CONNECTION	RRINDOUBT
DATASET	BACKUPNONBWO
DATASET	DELEXITERROR
DATASET	DATSETFULL
DATASET	DEADLOCK
DATASET	FAILEDDBKOUT
DATASET	INDEXRECFULL
DATASET	LCKSTRUCFULL
DATASET	IOERROR
DATASET	OPENERERROR
RLSSERVER	COMMITFAIL
RLSSERVER	RRCOMMITFAIL
RLSSERVER	RLSGONE
UNDEFINED	NOTAPPLIC

RLSACCESS (*cvda*)

Returns a CVDA value that indicates whether the data set was last opened in this CICS region in RLS or non-RLS mode. CVDA values are as follows:

NOTRLS

The last open in this CICS region was in non-RLS mode.

RLS The last open in this CICS region was in RLS mode.

SYSID (*data-area*)

Returns the 4-character sysid (when the CVDA on the CAUSE parameter is CONNECTION) of the remote system to which connectivity has been lost.

UOW (*data-area*)

Returns the 16-byte UOW identifier of a shunted unit of work that has one or more data sets with retained locks. The last eight bytes are always null (X'00').

Conditions**END**

RESP2 values:

2 There are no more UOW/data set pairs.

ILLOGIC

RESP2 values:

1 A START has been given when a browse is already in progress, or a NEXT has been given without a preceding START.

NOTAUTH

RESP2 values:

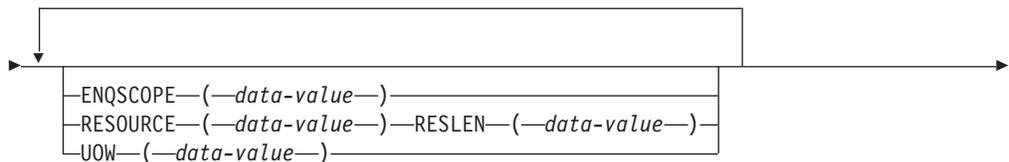
100 The use of this command is not authorized.

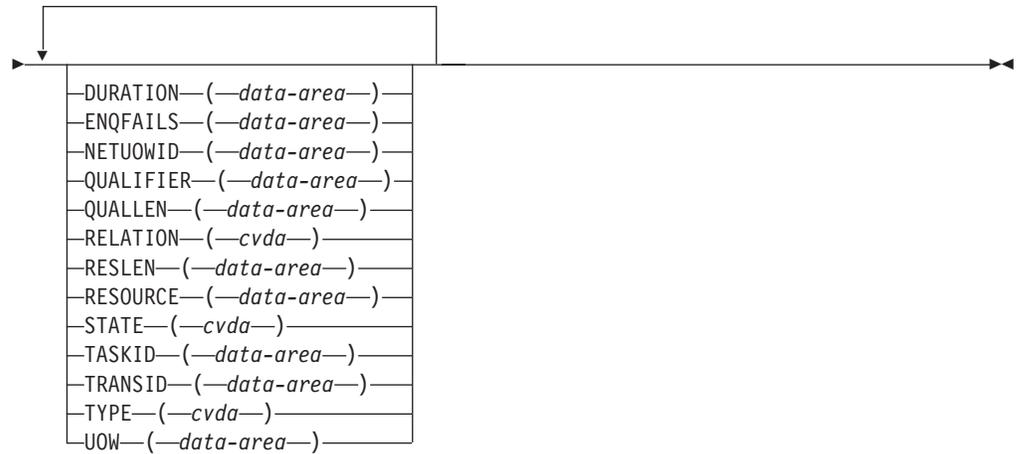
INQUIRE UOWENQ

Retrieve information about enqueues held or waited on by a UOW, or about UOWs holding or waiting on a specified enqueue. INQUIRE ENQ is a synonym for INQUIRE UOWENQ.

INQUIRE UOWENQ

▶▶ INQUIRE UOWENQ —————▶▶





Conditions: END, ILLOGIC, NOTAUTH, UOWNOTFOUND

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The INQUIRE UOWENQ command is for use only in browse mode and retrieves information about enqueues. CICS uses enqueues to lock recoverable resources, such as file records or queues, to the UOW that is updating them. User enqueues obtained by the EXEC CICS ENQ command are also returned.

The browse can be filtered in three ways:

- Supply a value for UOW on the START command to return only the enqueues held or waited on by the specified UOW.
- Supply a value for RESOURCE on the START command to return only information about UOWs owning or waiting on the specified enqueue.
- Supply a value for ENQSCOPE on the START command to return only enqueues with the specified enqscope. If ENQSCOPE is specified as blanks, only local enqueues are returned.

A CICS-wide browse occurs when you do not supply a value for UOW, RESOURCE or ENQSCOPE on the INQUIRE UOWENQ START command. All enqueue owners and enqueue waiters on the local system are returned by the browse. They are returned by considering each UOW in turn. After all the enqueues owned by one UOW have been returned, those owned by the next UOW in the system are considered.

As well as returning information about the owners of the enqueues, the command also Returns information about UOWs that are waiting on these enqueues. This enables you to diagnose enqueue deadlocks between tasks wanting to update the same resources. It provides a performance improvement over other methods of answering the question “Which UOW is holding the Enqueue?” when you want to analyze what the cause of a delay is.

Enqueues are typically held in active state, which means that other tasks are allowed to wait for the enqueue. However, if a UOW that owns enqueues suffers an indoubt failure, user ENQs are released while CICS enqueues are usually converted to the retained state until the indoubt failure can be resolved. User

ENQs are not to be used to lock recoverable resources, as they are not held across a CICS failure. The INQUIRE UOWENQ command also retrieves information about retained enqueues and can be used to identify which records and queues would be affected if the UOW were forced.

INQUIRE UOWENQ only Returns information about UOWs on the local system. For Enqueues with SYSPLEX SCOPE the OWNER may be on the local system with some or all of the waiters elsewhere, or the enqueue OWNER may be elsewhere in the sysplex with some or all of the waiters on the local system; In this case, only the local waiters are returned.

Browsing

Using the browse options (START, NEXT, and END) on INQUIRE UOWENQ commands, you can browse through all of the enqueues held by a specific UOW, or through all the enqueues currently in your system. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

The browse Returns both enqueue owners and enqueue waiters. They are returned by considering each UOW that owns an enqueue in turn. After all the enqueues owned by one UOW have been returned, those owned by the next UOW in the system are considered. Enqueue waiters are returned subsequent to the enqueue they are waiting on, but before the next enqueue owned by the current UOW. Note that the INQUIRE UOWENQ START does not retrieve data for the first enqueue. Also, because the enqueues are not returned in a defined order, you cannot specify a start point.

A CICS-wide browse occurs when you do not supply a value for UOW on the INQUIRE UOWENQ START command. All enqueue owners and waiters are returned by the browse. The first time an INQUIRE UOWENQ NEXT command is used, it Returns the data for the first enqueue that is owned. This is returned with RELATION(OWNER). If the enqueue has any waiters, the same enqueue is returned for each of these waiters, but this time with RELATION(WAITER). The UOW, NETUOWID, TASKID, and TRANSID fields each correspond to that particular waiter. All other data should be the same as when it was returned with RELATION(OWNER). After the last waiter has been returned, the next time the command is issued it Returns the next enqueue that is owned (if any).

If you supply a value for UOW on the START command, it acts as a “filter”, which means that only those enqueues owned by that particular UOW are returned (with a RELATION of OWNER). If the UOW happens to be waiting for an enqueue then this too is returned (but with a RELATION of WAITER).

Note that the enqueue state is not locked for the duration of the browse, or even between consecutive INQUIRE NEXT commands. To receive a consistent view of the state, the task performing the browse should not give up control to another task while the browse is in progress. If the owner of the last enqueue returned by the browse changes between successive INQUIRE NEXT commands, the browse Returns the enqueue again with its new owner and waiters.

Notes:

1. If there are many enqueues in the system, CICS may take a long time to process a browse. If this happens, consider increasing the runaway interval of tasks that perform browses. (Do this by increasing the value of the RUNAWAY attribute on the associated TRANSACTION definition).

2. Both UOW-lifetime and task-lifetime enqueues are returned by INQUIRE UOWENQ. (For an explanation of UOW- and task-lifetime enqueues, see the MAXLIFETIME option of the EXEC CICS ENQ command.)
3. On an indoubt failure, user enqueues are released, *unless* the EXEC CICS ENQ command specified MAXLIFETIME(TASK) and it is not the end-of-task syncpoint that suffers the failure.

Options

DURATION(*data-area*)

Returns, as a fullword value binary value, the elapsed time in seconds since the enqueue entered its current state of owner, waiter or retained.

ENQFAILS(*data-area*)

Returns, for retained enqueues, the number of failed enqueue attempts for this resource after the enqueue was last acquired. This indicates how many UOWs have received a LOCKED response because this enqueue was held in retained state. For active enqueues, ENQFAILS Returns zero.

Because the ENQFAILS option indicates how many UOWs are failing because of retained locks, you can use it to help identify which shunted UOWs are causing bottlenecks.

ENQSCOPE(*data-area*)

If the enqueue has sysplex scope, ENQSCOPE Returns the 4-character name which was used to qualify the sysplex-wide ENQUEUE request issued by this CICS region. If it has region scope, ENQSCOPE Returns blanks.

All CICS systems with the same ENQSCOPE value share the same sysplex Enqueue name space.

ENQSCOPE may also be used to supply a value on the START command. This limits the INQUIRE to return only enqueues with the specified scope name. If ENQSCOPE is specified as blanks, only local enqueues are returned.

NETUOWID(*data-area*)

Returns the 1- through 27-character network-wide LU6.2 ID of the UOW that owns or is waiting for the enqueue for which data is being returned.

QUALIFIER(*data-area*)

Returns a 0- through 255-character optional qualifier that further identifies the resource associated with the enqueue. The data (if any) returned in this field depends on the TYPE of the enqueue, as summarized in Table 5 on page 549.

QUALLEN(*data-area*)

Returns a halfword binary value indicating the length of the data, in the range 0 through 255, returned in the QUALIFIER field. If no QUALIFIER data is applicable to the resource (that is, for EXECQENQ, EXECENQADDR, and TSQUEUE), a value of zero is returned.

RELATION(*cvda*)

Returns a CVDA value indicating whether the data being returned is associated with the owner of the enqueue or with a task waiting for the enqueue. CVDA values are:

OWNER

The UOW, NETUOWID, TASKID, and TRANSID are those of the owner of the enqueue.

WAITER

The UOW, NETUOWID, TASKID, and TRANSID are those of a waiter for the enqueue.

RESLEN(*data-area*)

Returns a halfword binary value indicating the length of the data, in the range 1 through 255, returned in the RESOURCE field.

If RESOURCE is used as input on a START command, a RESLEN input is also required.

RESOURCE(*data-area*)

Returns the 1- through 255-character name of the resource associated with the enqueue lock. The data returned in this field depends on the TYPE of the enqueue, as summarized in Table 5 on page 549.

RESOURCE may also be used to supply a value on the START command. This limits the INQUIRE to return only information about UOWs owning or waiting on the specified enqueue.

STATE(*cvda*)

Returns a CVDA value indicating the state that the enqueue being returned is held in. It is returned on the INQUIRE UOWENQ NEXT command. CVDA values are:

ACTIVE

The enqueue is held in active state.

RETAINED

The enqueue is held in retained state. Its owning UOW has been shunted, or is in the process of being shunted.

TASKID(*data-area*)

Returns a 4-byte packed-decimal value giving the number of the task associated with the UOW. If the UOW is shunted, this is the task number associated with the UOW before it was shunted.

TRANSID(*data-area*)

Returns the 1- through 4-character identifier of the transaction associated with the UOW. If the UOW is shunted, it is the identifier of the transaction associated with the UOW before it was shunted.

TYPE(*cvda*)

Returns a CVDA value identifying the type of resource being enqueued upon. CVDA values are:

DATASET

The resource is a record in a VSAM data set opened in non-RLS mode (or a CICS-maintained data table). RESOURCE contains the name of the data set, and QUALIFIER contains the record identifier. Note that CICS does not hold enqueues on non-RLS data sets opened in RLS mode; in this case VSAM does the locking.

EXECENQ

The resource is associated with an EXEC CICS ENQ request. RESOURCE contains the enqueue argument passed on the request.

EXECENQADDR

The resource is associated with an EXEC CICS ENQ request. RESOURCE contains the address enqueue argument passed on the request (that is, the LENGTH parameter was omitted on the request)

FILE

The resource is a record in either a BDAM file or a user-maintained data table. RESOURCE contains the name of the file and QUALIFIER contains the record identifier.

When the file is a BDAM file then the record identifier is prefixed by the BDAM block identifier. Note that truncation occurs if this combination exceeds 255 characters.

TDQUEUE

The resource is a logically-recoverable transient data queue. RESOURCE contains the name of the queue. QUALIFIER contains either the string "FROMQ" or "TOQ", indicating whether an input or output lock is held for that queue.

Note that the definition of the WAITACTION attribute on the TDQUEUE resource definition determines what happens to TDQUEUE enqueues on an indoubt failure. For information on defining the WAITACTION attribute, see the *CICS Resource Definition Guide*.

A READQ TD request acquires the "FROMQ" lock, whereas a WRITEQ TD request acquires the "TOQ" lock associated with the queue. A DELETEQ TD request acquires both the "TOQ" and the "FROMQ" locks.

TSQUEUE

The resource is a recoverable temporary storage queue. RESOURCE contains the name of the queue.

Unlike other components, enqueues associated with recoverable temporary storage queues are only ever the retained kind; owned by a UOW that has been shunted as a result of an indoubt failure. The temporary storage component uses its own mechanism for locking queues to in-flight UOWs.

The data returned in the RESOURCE and QUALIFIER fields depends on the resource TYPE, as shown in Table 5.

Table 5. Data returned in RESOURCE and QUALIFIER

TYPE	RESOURCE	QUALIFIER
DATASET	Data set name	Record identifier
EXECENQ	EXEC enqueue argument	None
EXECENQADDR	Address of EXEC enqueue argument	None
FILE	File name	Record identifier
TDQUEUE	TD queue name	FROMQ or TOQ
TSQUEUE	TS queue name	None

UOW(*data-area*)

Returns the 16-byte local identifier of the UOW that owns or is waiting for the enqueue for which data is being returned. The last eight bytes are always null (X'00').

The UOW field may also be used to supply a value on the START command. This limits the INQUIRE to return only the enqueues held or waited on by the specified UOW.

Conditions

END

RESP2 values:

- 2 All enqueues have been retrieved.

Browsing

You can browse through all UOW links by using the browse options (START, NEXT, and END) on INQUIRE UOWLINK commands. See “Browsing resource definitions” on page 20 for general information about browsing, including syntax, exception conditions, and examples.

The browse form of the command returns the state of all the UOW links for connections that are *directly* connected to the CICS from which the command is issued. It indicates which connections are unavailable, or have been cold started.

The INQUIRE UOWLINK START command positions an internal pointer at the first UOW link in the CICS recovery manager table. It does not retrieve data for the first one, and it does not allow you to specify a start point.

The first time an INQUIRE UOWLINK NEXT command is used, it returns information about the first UOW link. Each time the command is used again, it retrieves the ID and STATE of the next UOW link, if one exists. You can filter the UOW-links returned by specifying a value in the UOW field.

The browse guarantees that data for each UOW link that exists before the first INQUIRE NEXT, and still exists after the last INQUIRE NEXT, is returned on exactly one INQUIRE NEXT call.

Options

HOST(*data-area*)

For TYPE values IIOP or IPIC, returns the TCP/IP hostname, or a string containing the colon hexadecimal or dotted decimal TCP/IP address, used to refer to the participant in the OTS transaction. This name is useful for identifying the participant, especially when problems occur. HOST is a 255-character data-area. Strings of fewer than 255 characters are padded with blanks. The UOW and the participant can belong to the same CorbaServer.

For TYPE values other than IIOP, HOST returns blanks.

LINK(*data-area*)

The value returned depends on the TYPE of connection that is returned:

RMI

The entry name of the task-related user exit.

CFTABLE

The 8-character name of the coupling facility data table pool

IIOP

Blanks. The equivalent IIOP information is returned by the HOST attribute.

IPCONN

The 8-character APPLID of the remote system.

NETUOWID(*data-area*)

Returns the 1- 27-character network-wide LU6.2 ID of the UOW for which data is returned.

PROTOCOL(*cvda*)

Returns a CVDA value indicating the communication protocol used by the connection. CVDA values are as follows:

APPC Advanced Program to Program Communication.

- IRC** Interregion Communication. This connection is an MRO connection.
- LU61** LUTYPE 6.1.
- IPIC** IP interconnectivity. This type of connection is made using an IPCONN resource.
- NOTAPPLIC**
This connection is of type CFTABLE or RMI.
- OTS** An OTS link.
- RRMS**
The UOW is coordinated by RRS/MVS.

RESYNCSTATUS(*cvda*)

Returns a CVDA value indicating the resynchronization status of the connection. CVDA values are as follows:

- COLD** The connection has been cold started by the partner system. The partner can no longer coordinate any indoubt conditions for this system; nor can this system pass to the partner any decisions remembered for it.

NOTAPPLIC

The connection was not created using recovery protocols. It might be an RMI, an APPC single-session, an APPC sync level 1 connection, an IPCONN, or a CFDT server.

- OK** The connection is operating normally. If the partner system has failed, the partner has been restarted and the connection is able to resynchronize the associated UOW.

STARTING

The connection is being acquired, but the exchange lognames process has not yet completed.

UNAVAILABLE

The connection is not currently acquired.

UNCONNECTED

No associated connection.

RMIQFY(*data-area*)

Returns, for a TYPE value of RMI, the 8-character entry qualifier of the task-related user exit. For a TYPE of CONNECTION, IPCONN, CFTABLE or IIOP, RMIQFY returns blanks.

ROLE(*cvda*)

Returns a CVDA value indicating the role of the connection. CVDA values are:

COORDINATOR

This connection is to the sync point coordinator for the UOW.

SUBORDINATE

This connection is to a sync point subordinate for the UOW.

UNKNOWN

The sync point role of this connection cannot be determined.

SYSID(*data area*)

Returns, for a TYPE value of CONNECTION, the 4-character SYSID of the connection. If the connection has been discarded, or the type is RMS, CFTABLE, IPCONN, or IIOP, or the PROTOCOL option returns RRMS, SYSID returns blanks.

TYPE(*cvda*)

Returns a CVDA value indicating the type of connection. CVDA values are as follows:

CFTABLE

A connection to a CFDT server.

CONNECTION

A connection defined in a CONNECTION resource definition.

IIOP OTS.**IPCONN**

A connection defined in an IPCONN resource definition.

RMI

A connection to an external resource manager using the resource manager interface (RMI).

UOW(*data-area*)

Returns the 16-byte local identifier of the UOW for which link data is being returned. The last eight bytes are always null (X'00').

UOWLINK(*data-area*)

Specifies a 4-byte token identifying the UOW-link for which data is to be returned.

URID(*data-area*)

If the PROTOCOL field returns RRMS, this option returns the 32-byte hexadecimal representation of the RRMS unit of recovery identifier. For other values of PROTOCOL, including OTS, URID returns blanks.

XID(*data-area*)

Returns the 64-character global transaction identifier, if present.

Conditions**END**

RESP2 values:

2 All authorized resource definitions have been retrieved.

ILLOGIC

RESP2 values:

1 For INQUIRE UOWLINK START, means that a browse of this resource type is already in progress. For INQUIRE UOWLINK NEXT and INQUIRE UOWLINK END, means that an INQUIRE UOWLINK START command has not been issued.

NOTAUTH

RESP2 values:

100 The use of this command is not authorized.

UOWLNOTFOUND

RESP2 values:

1 The named UOW-link cannot be found.

UOWNOTFOUND

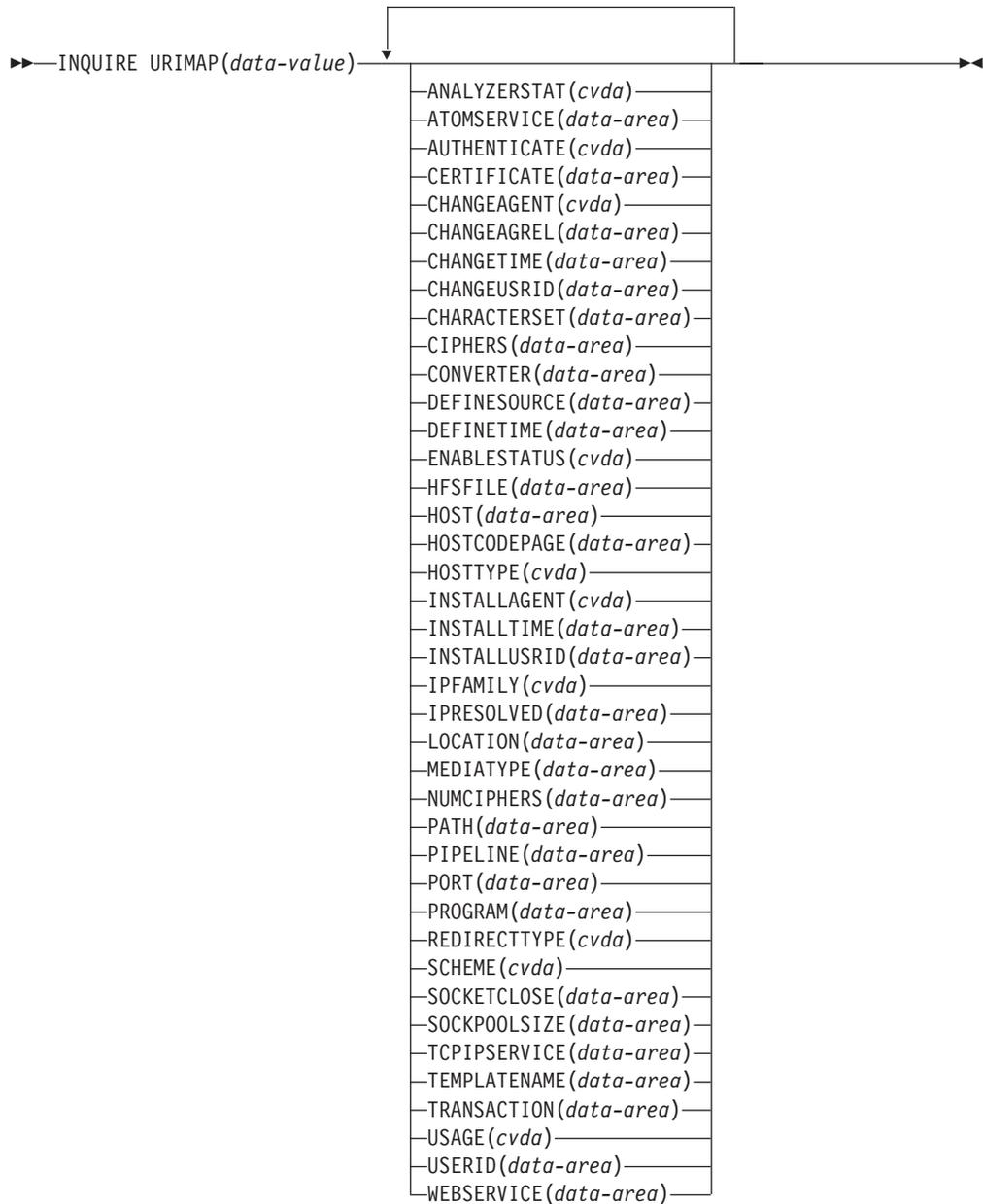
RESP2 values:

1 The named UOW cannot be found.

INQUIRE URIMAP

Retrieve information about URIMAP resources in the local system.

INQUIRE URIMAP



Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.

Browsing

You can also browse through all the URIMAP definitions installed in the region, using the browse options (START, NEXT, and END) on INQUIRE URIMAP commands. See Browsing resource definitions in CICS System Programming

Reference for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID, DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

ANALYZERSTAT (*cvda*)

Returns a CVDA value indicating whether the analyzer program associated with the TCPIPSERVICE definition is to be run. CVDA values are as follows:

ANALYZER

The analyzer program is to be run.

NOANALYZER

The analyzer program is not to be run.

This attribute is for USAGE(SERVER). For all other usage types, it is forced to NO.

ATOMSERVICE (*data-area*)

Returns the 1- to 8-character name of an ATOMSERVICE resource definition for an Atom feed. The ATOMSERVICE resource definition defines an Atom service, feed, collection, or category document, and identifies the Atom configuration file, CICS resource or application program, and XML binding that are used to supply the data for the feed. This attribute is for USAGE(ATOM).

AUTHENTICATE (*cvda*)

Returns a CVDA value indicating whether to provide authentication information to a web services provider. CVDA values are as follows:

BASICAUTH

The web services provider requires HTTP basic authentication. You can supply credentials to the web services requester (a user ID and password) to the global user exit, XWBAUTH, which, if enabled, sends the credentials to the web services provider.

NOAUTHENTIC

The web services provider does not require authentication.

CERTIFICATE (*data-area*)

Returns a 32-character data area, which contains the label of the certificate that is to be used as the SSL client certificate for the HTTP request by CICS as an HTTP client. This attribute is for USAGE(CLIENT).

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

DYNAMIC

The resource definition was last changed by a PIPELINE scan.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

CHARACTERSET (*data-area*)

Returns a 40-character data area containing the name of the character set to be used for the static response. This attribute is for **USAGE(SERVER)**.

CIPHERS (*data-area*)

Returns a 56-character data area containing the list of cipher suites specified for the URIMAP definition. The list of cipher suites is used to negotiate SSL connections. This attribute is for **USAGE(CLIENT)**.

CONVERTER (*data-area*)

Returns the 8-character name of a converter program that performs conversion or other processing for CICS as an HTTP server. This attribute is for **USAGE(SERVER)**.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT** value. For more information, see **Summary of the resource signature field values in the Resource Definition Guide**.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

ENABLESTATUS (*cvda*)

Returns a CVDA value indicating the status of this URIMAP definition. CVDA values are as follows:

ENABLED

The URIMAP definition is enabled.

DISABLED

The URIMAP definition is disabled. A URIMAP definition with this status can be discarded.

DISABLEDHOST

The URIMAP definition is unavailable because the virtual host of which it is a part has been disabled. The SET HOST command can be used to reenable all the URIMAP definitions that make up the virtual host. A URIMAP definition with this status cannot be discarded.

HFSFILE (*data-area*)

Returns a 255-character data area containing fully qualified (absolute) or relative name of a z/OS UNIX System Services file that forms a static response. This attribute is for USAGE(SERVER).

HOST (*data-area*)

Returns the 116-character host name or its IPv4 or IPv6 address. The HOST option can be a character host name, an IPv4 address, or an IPv6 address. HOST is specified in the resource definition. HOST displays all IPv4 addresses as native IPv4 dotted decimal addresses, for example, 1.2.3.4, regardless of the type of address format used. You can specify IPv4 and IPv6 addresses in a number of acceptable formats. See IP addresses in the Internet Guide for more information about address formats.

For USAGE(CLIENT), the port number is also displayed in the HOST option if HOST contains a native IPv4 address or a host name; however, if you specify a hostname that is greater than 110 characters in length, port information is not displayed in the HOST option. This rule also applies if you specify an IPv4 address in IPv6 format. Use the PORT option to view the port number.

HOSTCODEPAGE (*data-area*)

Returns a 10-character data area containing the 1- to 10-character name of the IBM code page (EBCDIC) in which the text document that forms the static response is encoded. This attribute is for USAGE(SERVER).

HOSTTYPE (*cvda*)

Returns the address format of the HOST option. HOSTTYPE is set by CICS when the URIMAP is installed. CVDA values are as follows:

HOSTNAME

The HOST option contains a character host name. The IP address that corresponds to the host name is looked up in the domain name server.

IPV4 The address is specified in IPv4 dotted decimal address format.

IPV6 The address is specified in IPv6 colon hexadecimal address format.

NOTAPPLIC

An incorrect host address was returned (HOST=0.0.0.0 or HOST=*), or the HOSTTYPE option is used with URIMAP(ATOM), URIMAP(PIPELINE), or URIMAP(SERVER).

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

BUNDLE

The resource was installed by a bundle deployment.

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

DYNAMIC

The resource was installed by using a PIPELINE scan.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID (*data-area*)

Returns the 8-character user ID that installed the resource.

IPFAMILY (*cvda*)

Returns the address format of the IPRESOLVED option. CVDA values are as follows:

IPV4 The address is specified in IPv4 dotted decimal address format.

IPV6 The address is specified in IPv6 colon hexadecimal address format.

UNKNOWN

IPRESOLVED is not yet in use or the address cannot be resolved.

UNKNOWN is the default when IPRESOLVED is 0.0.0.0, or if the IPFAMILY option is used with URIMAP(ATOM), URIMAP(PIPELINE), or URIMAP(SERVER).

IPRESOLVED (*data-area*)

Returns a 39-character field that specifies the IPv4 or IPv6 address of the HOST option. This attribute is for all types except USAGE(SERVER). If the URIMAP is installed but has not yet been used to establish a connection, or the address cannot be resolved, a default value of 0.0.0.0 is returned. When the URIMAP establishes a connection, IPRESOLVED displays the resolved IP address that was used by the resource to connect. IPRESOLVED is reset to 0.0.0.0 when the resource is disabled. The content of IPRESOLVED is not recoverable after a warm or emergency restart.

LOCATION (*data-area*)

Returns a 255-character area containing a URL to which matching HTTP requests from Web clients are redirected. Redirection is activated by the setting specified by the REDIRECTTYPE option. This attribute is for USAGE(SERVER), USAGE(PIPELINE), or USAGE(ATOM).

MEDIATYPE (*data-area*)

Returns a 56-character data area containing a description of the data content of the static response. This attribute is for USAGE(SERVER).

NUMCIPHERS (*data-area*)

Returns a halfword binary value containing the number of cipher codes in the CIPHERS list. The ciphers are used to negotiate encryption levels as part of the SSL handshake. This attribute is for USAGE(CLIENT).

PATH (*data-area*)

Returns a 255-character data area containing the path component of the URL to which the URIMAP definition applies; for example, software/htp/cics/index.html. This attribute is for any usage type.

PIPELINE (*data-area*)

Returns the 8-character name of the PIPELINE resource definition for the Web

service. The PIPELINE resource definition provides information about the message handlers that act on the service request from the client. This attribute is for USAGE(PIPELINE).

PORT(*data-area*)

Returns a fullword binary value, in the range 1 - 65535, containing the port number value of the connection to the server.

For USAGE(CLIENT), the PORT option displays the port number used for the client connection. The port number is also displayed in the HOST option if HOST contains a native IPv4 address or a host name. For USAGE(CLIENT), the PORT attribute always contains the port number that is being used for the communication, even if PORT(NO) is specified on the URIMAP at define time. PORT is specified in the resource definition.

For USAGE(ATOM), USAGE(SERVER), or USAGE(PIPELINE), the PORT option is set to PORT(NO).

PROGRAM(*data-area*)

Returns the 8-character name of the application program that composes an application-generated response to the HTTP request. This option is for USAGE(SERVER).

REDIRECTTYPE(*cvda*)

Returns a CVDA value indicating the type of redirection for requests that match this URIMAP definition. The URL for redirection is specified by the LOCATION option. This attribute is for USAGE(SERVER), USAGE(PIPELINE), or USAGE(ATOM). CVDA values are as follows:

NONE

Requests are not redirected. Any URL specified by the LOCATION option is ignored.

TEMPORARY

Requests are redirected on a temporary basis. The status code used for the response is 302 (Found).

PERMANENT

Requests are redirected permanently. The status code used for the response is 301 (Moved Permanently).

SCHEME(*cvda*)

Returns a CVDA value indicating the scheme component of the URI. CVDA values are as follows:

HTTP HTTP without SSL.

HTTPS

HTTP with SSL.

This attribute is for any usage type.

SOCKETCLOSE(*data-area*)

Returns, in fullword binary form, the maximum length of time in seconds that CICS keeps a client HTTP connection open for reuse after the CICS application has finished using it. If the value is 0, CICS does not keep connections open for reuse. This attribute is for USAGE(CLIENT). For other usage types, CICS returns a null value (-1).

SOCKPOOLSIZE(*data-area*)

Returns, in fullword binary form, the number of client HTTP connections that CICS is currently holding in a pool in a dormant state. The connections can be

reused by any CICS application that connects as a Web client to the same host and port. This attribute is for USAGE(CLIENT). For other usage types, CICS returns a null value (-1).

TCPIPSERVICE(*data-area*)

Returns the 1- to 8-character name of the TCPIPSERVICE definition that specifies an inbound port to which this URIMAP definition relates. If this port is not specified, the URIMAP definition applies to a request on any inbound ports. This attribute is for USAGE(SERVER), USAGE(PIPELINE), or USAGE(ATOM).

TEMPLATENAME(*data-area*)

Returns a 48-character data area containing the name of a CICS document template that is used to form a static response. This attribute is for USAGE(SERVER).

TRANSACTION(*data-area*)

Returns the 4-character name of an alias transaction to run the user application that composes a response to the HTTP request. This attribute is for USAGE(SERVER), USAGE(PIPELINE), or USAGE(ATOM).

URIMAP(*data-value*)

Returns the 8-character name of a URIMAP definition.

USAGE(*cvda*)

Returns a CVDA value indicating the purpose of this URIMAP definition. CVDA values are as follows:

SERVER

A URIMAP definition for CICS as an HTTP server. This type of URIMAP definition is used to map the URL of an incoming HTTP request from a Web client to CICS resources. An application-generated response or a static response can be provided.

CLIENT

A URIMAP definition for CICS as an HTTP client. This type of URIMAP definition is used when CICS makes a client request for an HTTP resource on a server.

PIPELINE

A URIMAP definition for a Web service. This type of URIMAP definition is used to specify the processing that is to be performed on a request by which a client calls a Web service in CICS.

ATOM

A URIMAP definition for an Atom feed. This type of URIMAP definition is used for an incoming request for data that CICS makes available as an Atom feed. The URIMAP definition maps the request URI to an ATOMSERVICE resource definition, which defines an Atom document.

USERID(*data-area*)

Returns the 8-character user ID under which the alias transaction is attached. This attribute is for USAGE(SERVER), USAGE(PIPELINE), or USAGE(ATOM).

WEBSERVICE(*data-area*)

Returns the name of a Web service. This name can be the 1- to 8-character name of a WEBSERVICE resource definition or a name up to 32 characters representing a Web service generated by the CICS Web services assistant. This

attribute defines aspects of the runtime environment for a CICS application program deployed in a Web services setting. This attribute is for USAGE(PIPELINE).

Conditions

END

RESP2 values are:

- 2 There are no more resource definitions of this type.

ILLOGIC

RESP2 values are:

- 1 You have issued a START command when a browse of this resource type is already in progress, or you have issued a NEXT or an END command when a browse of this resource type is not in progress.

NOTAUTH

RESP2 values are:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values are:

- 3 The URIMAP cannot be found.

INQUIRE VOLUME

This command is supported in releases of CICS earlier than CICS TS for OS/390, for retrieving information about journal volumes.

INQUIRE VOLUME is obsolete, and is retained only for compatibility with previous releases. The only run-time support is to return the VOLIDERR condition. If this command is used, the translator translates it, but issues a warning message

The browse function is provided for compatibility with releases of CICS earlier than CICS TS for OS/390. A NORMAL condition is returned for the START browse and END browse operations. The ENDCOND condition is returned for the NEXT browse operation.

Conditions

VOLIDERR

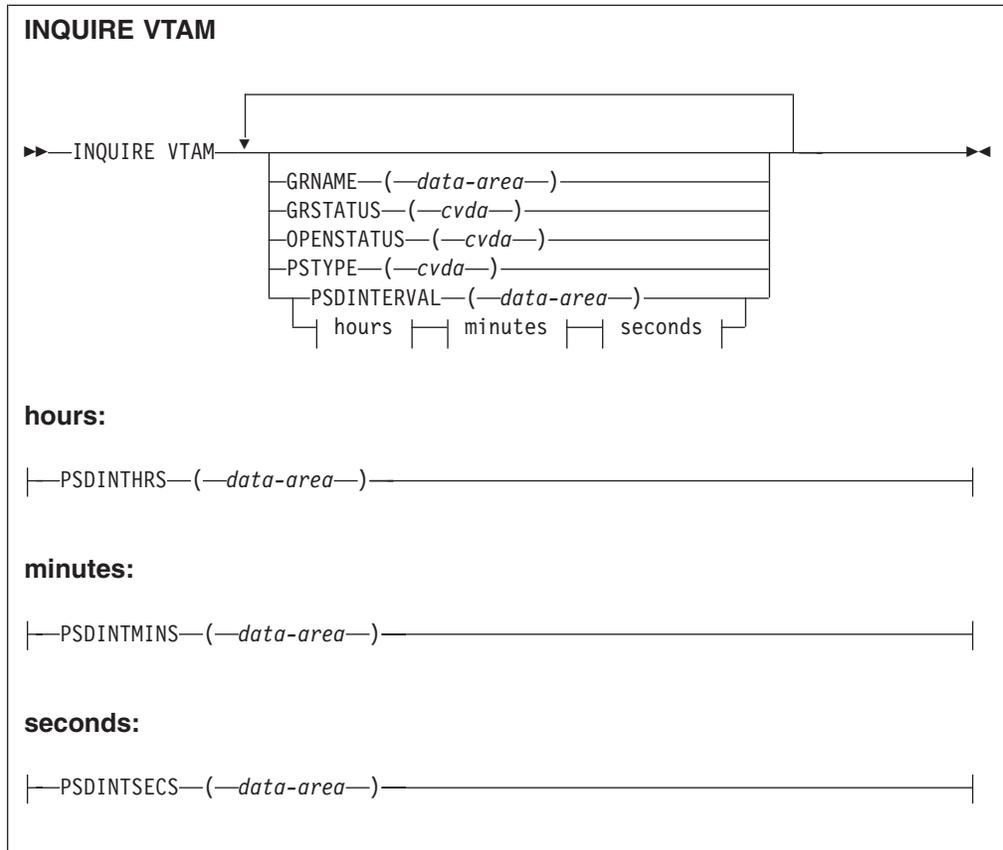
RESP2 values:

- 1 The program has issued an INQUIRE VOLUME browse command. This command is withdrawn.

INQUIRE VTAM

Retrieve information about the connection between CICS and the z/OS Communications Server.

Note: VTAM is now the z/OS Communications Server.



Conditions: INVREQ, NOTAUTH

For more information about the use of CVDAs, see CICS-value data areas (CVDAs) in CICS System Programming Reference.

Description

The INQUIRE VTAM command returns information about the type and state of the connection between the Communications Server and your CICS system.

Options

GRNAME (*data-area*)

Returns the 8-character generic resource group name under which this CICS region requests registration to the Communications Server, if it is using the generic resources facility of the Communications Server. Blanks are returned if the system was initialized without a request for registration.

GRSTATUS (*cvda*)

Returns a CVDA value indicating the status of generic resource registration. All of the values except NOTAPPLIC indicate that CICS has been initialized to use the generic resource function; that is, a nonblank GRNAME value was specified. CVDA values are:

DEREGERROR

Deregistration was attempted but was unsuccessful, and no attempt to reregister has been made.

DEREGISTERED

Deregistration was successfully accomplished.

NOTAPPLIC

CICS is not using the generic resource feature; GRNAME is not set or is set to blanks.

REGERROR

Registration was attempted but was unsuccessful, and no attempt to unregister has been made.

REGISTERED

Registration was successful and no attempt to unregister has been made.

UNAVAILABLE

The Communications Server does not support the generic resource function.

UNREGISTERED

CICS is using the generic resource function but no attempt, as yet, has been made to register.

OPENSTATUS (*cvda*)

Returns a CVDA value indicating the status of the connection between CICS and the Communications Server. CVDA values are:

CLOSED

The connection between CICS and the Communications Server has not yet been established or has been stopped.

CLOSEFAILED

The connection is open but is not usable because a previous request to close the connection failed. Retry the close request.

CLOSING

The connection between CICS and the Communications Server is in the process of closing.

FORCECLOSING

The connection between CICS and the Communications Server is in the process of closing following a SET VTAM FORCECLOSE command.

IMMCLOSING

The connection between CICS and the Communications Server is in the process of closing following a SET VTAM IMMCLOSE command.

OPEN A connection exists between CICS and the Communications Server.

PSDINTERVAL (*data-area*)

Returns the persistent session delay interval, which is the length of time that Communications Server sessions are held in recovery-pending state after a failure. The **PSDINT** system initialization parameter specifies this value for the CICS region at startup. The persistent session delay interval has two formats:

- A composite (packed decimal format *0hhmmss+*, 4 bytes long), which you obtain by using the PSDINTERVAL option.
- Separate hours, minutes, and seconds, which you obtain by specifying the PSDINTHRS, PSDINTMINS, and PSDINTSECS options.

A value of zero means that sessions are not retained after a failure.

PSDINTHRS (*data-area*)

Returns the hours component of the persistent session delay interval, in fullword binary form. See the PSDINTERVAL option.

PSDINTMINS (*data-area*)

Returns the minutes component of the persistent session delay interval, in fullword binary form. See the PSDINTERVAL option.

PSDINTSECS (*data-area*)

Returns the seconds component of the persistent session delay interval, in fullword binary form. See the PSDINTERVAL option.

PSTYPE (*cvda*)

Returns a CVDA value indicating the type of Communications Server persistent sessions support for the CICS region. CVDA values are as follows:

SNPS Single-node persistent sessions. Communications Server sessions can be recovered after a CICS failure and restart.

MNPS

Multinode persistent sessions. Communications Server sessions can also be recovered after a Communications Server or z/OS failure in a sysplex.

NOPS Communications Server persistent sessions support is not used for this CICS region.

Conditions**INVREQ**

RESP2 values:

1 The Communications Server is not present in the system.

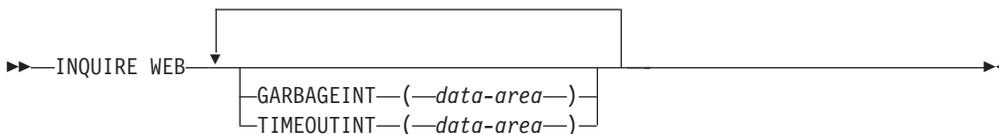
NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

INQUIRE WEB

Retrieve information about CICS Web support.

INQUIRE WEB

Conditions: NOTAUTH

This command is threadsafe.

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The **INQUIRE WEB** command returns information about the status of the CICS Web interface.

Options

GARBAGEINT (*data-area*)

Returns, in fullword binary form, the interval, in minutes, at which the Web garbage collection task runs to clean up Web 3270 state data for which the terminal timeout interval has expired.

TIMEOUTINT (*data-area*)

Returns, in fullword binary form, the time, in minutes, after which inactive Web 3270 sessions are eligible for garbage collection.

Conditions

NOTAUTH

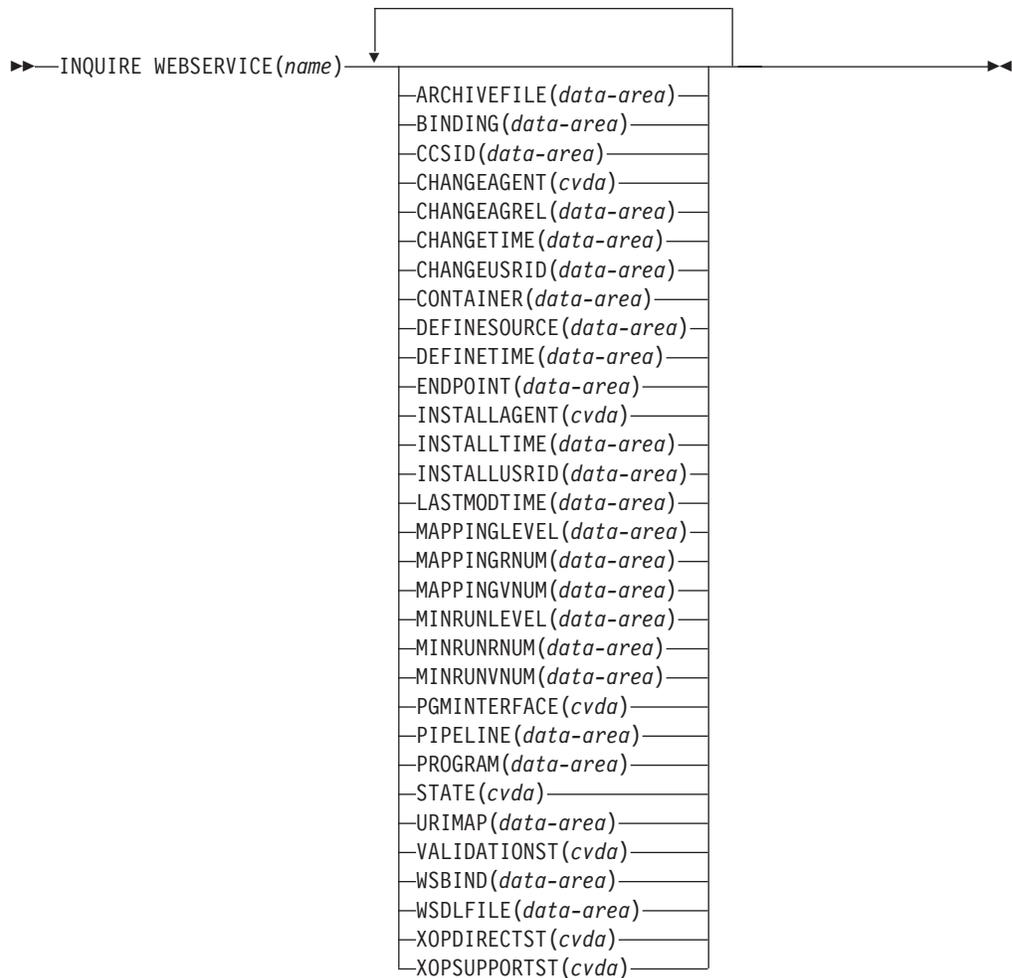
RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

INQUIRE WEBSERVICE

Use the **INQUIRE WEBSERVICE** command to retrieve information about an installed web service.

INQUIRE WEBSERVICE



Conditions: NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the **INQUIRE WEBSERVICE** command to retrieve information about an installed web service.

Browsing

You can browse through all the web services installed in your system by using the browse options, **START**, **NEXT**, and **END**, on **INQUIRE WEBSERVICE** commands. See Browsing resource definitions in CICS System Programming Reference for general information about browsing, including syntax, exception conditions, and examples.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are **CHANGEAGENT**, **CHANGEAGREL**, **CHANGETIME**, **CHANGEUSRID**,

DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

ARCHIVEFILE (*data-area*)

Returns the name of an archive file that contains one or more WSDL files. The name can be up to 255 characters in length.

BINDING (*data-area*)

Returns the WSDL binding represented by the WEBSERVICE resource. This binding is one of (potentially) many that appear in the WSDL file. The name can be up to 255 characters long.

CCSID (*data-area*)

Returns the CCSID that is used to encode the character data in the application data structure at run time. This value is set using the optional **CCSID** parameter in the web services assistant when the web service binding file was generated. If the *data-area* is 0, the default CCSID for the CICS region that is specified by the **LOCALCCSID** system initialization parameter is used.

CHANGEAGENT (*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

DYNAMIC

The resource definition was last changed by a PIPELINE scan.

CHANGEAGREL (*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID (*data-area*)

Returns the 8-character user ID that ran the change agent.

CONTAINER (*data-area*)

Returns the name of the container used if the **PGMINTERFACE** option returns a value of **CHANNEL**. The name can be up to 16 characters long.

DEFINESOURCE (*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE**

value depends on the CHANGEAGENT value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

ENDPOINT (*data-area*)

Returns the endpoint URI of a remote web service. This endpoint URI is specified in the WSDL file for a remote web service. If a CICS application program is the service provider, the ENDPOINT is empty. The URI can be up to 255 characters long.

INSTALLAGENT (*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

BUNDLE

The resource was installed by a bundle deployment.

CREATESPI

The resource was installed by an **EXEC CICS CREATE** command.

CSDAPI

The resource was installed by a CEDA transaction or the programmable interface to DFHEDAP.

DYNAMIC

The resource was installed by using a PIPELINE scan.

GRPLIST

The resource was installed by **GRPLIST INSTALL**.

INSTALLTIME (*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID (*data-area*)

Returns the 8-character user ID that installed the resource.

LASTMODTIME (*data-area*)

Returns the time, in milliseconds since 00:00 on January 1st 1900, that the deployed WSBIND file on z/OS UNIX was last updated. This value is a readonly value that CICS updates when the WEBSERVICE resource is installed or updated. The last-modified-time can be used to determine whether CICS has refreshed itself after an update is made to a WSBIND file in the pickup directory.

- For dynamically-installed web services (those installed by the CICS scanning mechanism), the value of LASTMODTIME is the timestamp of the z/OS UNIX file pointed to by the WSBIND definition, at the time the WEBSERVICE definition was last installed or updated.
- For statically installed web services (those installed from a CSD or by CREATE WEBSERVICE), the value of LASTMODTIME is the timestamp of the WSBIND z/OS UNIX file pointed to by the WEBSERVICE definition, at the time that the web service was installed.

If you issue an **INQUIRE WEBSERVICE** command before a newly installed or updated web service has fully initialized, the returned LASTMODTIME value will be zero.

The value is returned in 8-byte packed-decimal form. You can use the **EXEC CICS FORMATTIME** command to convert the LASTMODTIME value to the date-and-time format that you prefer.

MAPPINGLEVEL(*data-area*)

Returns an 8-byte character string of the mapping level that is used to convert data between language structures and web service description (WSDL) documents. The value of the mapping level is 1.0, 1.1, 1.2, 2.0, or 2.1.

MAPPINGNUM(*data-area*)

Returns a fullword binary value of the release number for the mapping level that is used to convert data between language structures and web service description (WSDL) documents. The value of the release number is 0, 1, or 2.

MAPPINGVNUM(*data-area*)

Returns a fullword binary value of the version number for the mapping level that is used to convert data between language structures and web service description (WSDL) documents. The value of the version number is 1 or 2.

MINRUNLEVEL(*data-area*)

Returns an 8-byte character string of the minimum runtime level that is required to run the web service in CICS. The value of the runtime level is 1.0, 1.1, 1.2, 2.0, or 2.1.

MINRUNNUM(*data-area*)

Returns a fullword binary value of the release number for the minimum runtime level that is required to run the web service in CICS. The value of the release number is 0, 1, or 2.

MINRUNVNUM(*data-area*)

Returns a fullword binary value of the version number for the minimum runtime level that is required to run the web service in CICS. The value of the version number is 1 or 2.

PGMINTERFACE(*cvda*)

Returns a CVDA indicating whether the CICS program that implements the web service expects input in a channel or in a COMMAREA. CDVA values are as follows:

CHANNEL

The program expects input in a channel.

COMMAREA

The program expects input in a COMMAREA.

NOTAPPLIC

PGMINTERFACE does not apply when the web service is a service requester.

PIPELINE(*data-area*)

Returns the name of the PIPELINE resource in which the web service is installed; that is, the name of the PIPELINE resource that contains this WEBSERVICE resource. The name can be up to 8 characters long.

PROGRAM(*data-area*)

Returns the name of a CICS program that implements the web service. If this WEBSERVICE resource represents a remote web service (that is, CICS is not the service provider), the PROGRAM option is empty. The name can be up to 8 characters long.

STATE(*cvda*)

Returns a CVDA indicating the state of the web service:

DISCARDING

A DISCARD command has been issued for the WEBSERVICE resource. The WEBSERVICE resource is quiescing before being discarded. It is not accepting new work, but is allowing current work to complete.

INITING

The web service binding file and the WSDL file are being copied to the shelf.

INSERVICE

Resolution of the copy of the WSBIND file on the shelf has succeeded, and the web service is usable.

UNUSABLE

Copying of the WSBIND file on the shelf has failed, and the WEBSERVICE resource is unusable.

URIMAP(*data-area*)

Returns the name of a dynamically installed URIMAP definition if one is associated with this web service. If the web service was not installed by performing the SCAN function on a PIPELINE resource, or if the WEBSERVICE resource represents a remote web service, the URIMAP definition is empty. The name can be up to 8 characters long.

VALIDATIONST(*cvda*)

Returns a CVDA indicating whether full validation of SOAP messages is currently enabled for this web service. CDVA values are as follows:

VALIDATION

Full validation is enabled.

NOVALIDATION

Full validation is disabled.

WEBSERVICE(*name*)

Specifies the name of the web service about which you are inquiring. The name can be up to 32 characters long.

WSBIND(*data-area*)

Returns the name of the web service binding file. The name can be up to 255 characters long.

WSDLFILE(*data-area*)

Returns the name of the web service description file associated with the WEBSERVICE resource. The name can be up to 255 characters long.

XOPDIRECTST(*cvda*)

Returns a value that indicates whether the web service is currently able to handle XOP documents in direct mode. CDVA values are as follows:

NOXOPDIRECT

The web service cannot currently handle XOP documents and binary attachments directly. This value is true when the web service implementation does not support the direct handling of XOP documents and binary attachments, or web service validation is switched on.

XOPDIRECT

The web service can currently handle XOP documents and binary attachments directly. This value is true when the web service implementation supports the direct handling of XOP documents and web service validation is not switched on.

XOPSUPPORTST(*cvda*)

Returns a CVDA value that indicates whether the web service implementation can handle XOP documents and binary attachments in direct mode. The CVDA values are as follows:

NOXOPSUPPORT

The web service implementation does not support the direct handling of XOP documents and binary attachments.

XOPSUPPORT

The web service implementation supports the direct handling of XOP documents and binary attachments. This value is true for any web services that are generated and deployed using the web services assistant.

Conditions

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

- 3 The web service cannot be found.

INQUIRE WORKREQUEST

Retrieve information about work requests in the local CICS region.

INQUIRE WORKREQUEST

►►—INQUIRE WORKREQUEST—(—*data-value*—)—————►

—WORKTYPE—(— <i>cvda</i> —)—————
—CLIENTIPADDR—(— <i>data-area</i> —)——
—CLNTIP6ADDR—(— <i>data-area</i> —)——
—CLNTIPFAMILY—(— <i>cvda</i> —)————
—CORBASERVER—(— <i>data-area</i> —)——
—LISTENERPORT—(— <i>data-area</i> —)——
—OAPPLID—(— <i>data-area</i> —)————
—OTASK—(— <i>data-area</i> —)—————
—OTRANSID—(— <i>data-area</i> —)————
—REQUESTID—(— <i>data-area</i> —)——
—STACK—(— <i>data-area</i> —)—————
—TARGETSYS—(— <i>data-area</i> —)——
—TASK—(— <i>data-area</i> —)—————
—TRANSID—(— <i>data-area</i> —)————
—TSYSTEM—(— <i>data-area</i> —)————
—TSYSTYPE—(— <i>cvda</i> —)—————

Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.

For more information about the use of CVDAs, see CICS-value data areas (CVDA) in CICS System Programming Reference.

Description

Use the **INQUIRE WORKREQUEST** command to track tasks that are started as a result of action by a request receiver.

A listener region in CICS is an example of a RequestReceiver. A single RequestReceiver can start one RequestProcessor, which in turn can start more RequestReceivers or RequestProcessors.

The RequestReceiver and RequestProcessor relationship is not the same as any other type of transaction relationship. A RequestReceiver does not remain in the system after it has initiated a RequestProcessor; if there are no more requests to be received, it ends. A RequestProcessor can create RequestReceivers during the course of its processing, both outbound and loopback requests.

Use the **INQUIRE WORKREQUEST** command to inquire about a single work request in the local region or browse through all the work requests in the local region. You can use the information returned by **INQUIRE WORKREQUEST** in these ways:

- Determine which transactions are associated with this work request.
- Correlate all transactions associated with this work request; for example, for accounting purposes.

Not all fields are supported for all types of work request. For example, fields that are used to display IIOPI information return blanks for SOAP work requests. If you do not set a value for the **WORKTYPE** option, CICS uses IIOPI as the default value.

Work Requests are identified by tokens. Each token is unique in the local system for the lifetime of the request, but, subsequently, tokens can be reused.

Browsing

You can use the browse options, **START**, **NEXT**, and **END**, on **INQUIRE WORKREQUEST** commands, to browse through all of the work requests present in the local region. In browse mode, CICS uses a default value of **ALL** for the work type if you do not set a value for the **WORKTYPE** option. See Browsing resource definitions in CICS System Programming Reference for general information about browsing, including syntax, exception conditions, and examples.

The work requests are not returned in a defined order, so you cannot use **START AT** to specify a start point.

Options

CLIENTIPADDR(*data-area*)

Returns, into a 15-character area, the dotted decimal IPv4 address of the TCP/IP client that originated the request. Do not use **CLIENTIPADDR** for new programs; use **CLNTIP6ADDR** instead. If the client address is in IPv6 format, it is returned to **CLNTIP6ADDR**, and **CLIENTIPADDR** returns 0.0.0.0.

CLNTIP6ADDR(*data-area*)

Returns, into a 39-character area, the colon hexadecimal IPv6 address of the TCP/IP client that originated the request. If the client address is in IPv4 format, the dotted decimal IPv4 address is returned.

CLNTIPFAMILY(*cvda*)

Returns a value indicating the form of TCP/IP addressing used by this task. Here are the values:

IPV4 The CLIENTIPADDR and CLIENTIP6ADDR options contain a dotted decimal IPv4 address.

IPV6 The CLNTIP6ADDR option contains a colon hexadecimal IPv6 address.

NOTAPPLIC

No entry is specified in the CLIENTIPADDR or CLNTIP6ADDR option.

CORBASERVER(*data-area*)

Specifies the name of the CorbaServer as a 4-character value.

LISTENERPORT(*data-area*)

Returns, in fullword binary form, the host port that received the request.

OAPPLID(*data-area*)

Specifies the applid of the originating task as an 8-character value.

OTASK(*data-area*)

Returns, in packed decimal form, the number of the task (RequestReceiver) that received the request.

OTRANSID(*data-area*)

Specifies as a 4-character value, the transaction ID that received the request.

REQUESTID(*data-area*)

Returns, in fullword binary form, the current request. A client can send more than one request at a time.

STACK(*data-area*)

Returns, in fullword binary, the stack level of this call.

Each time a request receiver creates a work request, it sets up a stack and records a value of '1'. Then, when the request receiver issues a method request, it increments the stack, and, each time that a bean in this "chain of requests" issues a method request to another bean, it increments this stack. One stack is created for each request receiver, and the stack applies across the sysplex. The stack is decremented each time a bean that has incremented it ends.

TARGETSYS(*data-area*)

Returns the applid of the target CICS system as a 50-byte field. This field can contain one of the following values:

- The colon hexadecimal TCP/IP address and port number of the target system, in the format [::a:b:c:d]:port where [] are X'BA' & X'BB' (code page 37).
- The dotted decimal TCP/IP address and port number of the target system.
- The z/OS Communications Server applid of the target system. The applid can be up to 8 characters followed by blanks.
- A string of blank characters indicating that the target system is not CICS over MRO or that the target system is not CICS over TCP/IP, or that an error has occurred.
- A different value, which CICS captures but does not translate.

The TARGETSYS option returns the same information as TSYSTEM, but can also return an IPv6 format address. If TARGETSYS returns an IPv4 address, this address is also returned in TSYSTEM; otherwise, TSYSTEM returns 0.0.0.0.

If you are using IPv6 connections, you must use the TARGETSYS option for your queries, instead of TSYSTEM.

TASK(*data-area*)

Returns, in packed decimal form, the number of the local task (RequestProcessor).

TRANSID(*data-area*)

Specifies the local 4-character transaction ID.

TSYSTEM(*data-area*)

Returns the APPLID of the target CICS system as a 21-byte field. Do not use TSYSTEM for new programs, use TARGETSYS instead. This field can contain one of the following values:

- The dotted decimal TCP/IP address and port number of the target system.
- The z/OS Communications Server APPLID of the target system. The APPLID can be up to eight characters followed by blanks.
- A string of blank characters indicating that the target system is not CICS over MRO, or that the target system is not CICS over TCP/IP, or that an error has occurred.
- A different value, which CICS captures but does not translate.

The TARGETSYS option returns the same information as TSYSTEM, but can also return an IPv6 format address. If TARGETSYS returns an IPv4 address, this address is also returned in TSYSTEM; otherwise, TSYSTEM returns 0.0.0.0:0.

If you are using IPv6 connections, you must use the TARGETSYS option for your queries, instead of TSYSTEM.

TSYSTYPE(*cvda*)

Returns a value indicating the form of TCP/IP addressing used by this task. CVDA values are as follows:

IPV4 The TSYSTEM and TARGETSYS options contain a dotted decimal IPv4 address and a port.

IPV6 The TARGETSYS option contains a colon hexadecimal IPv6 address and a port. This value is returned only if an IPv6 address is returned in the TARGETSYS option.

APPLID

The TSYSTEM and TARGETSYS options contain an APPLID.

NOTAPPLIC

No entry is specified in the TSYSTEM or TARGETSYS option. Blanks or a different value are returned.

WORKREQUEST(*data-value*)

Specifies, as an 8-character field, a token to identify the work request. Valid characters are the 16 hex digits 0 - 9, A - F. The token is generated by CICS. Normal usage begins with an **INQUIRE WORKREQUEST BROWSE** command, to identify the token that is of interest. Then the specific token can be used to issue INQUIRE and SET commands that relate to the task in question.

WORKTYPE(*cvda*)

Specifies the type of work that you want to inquire about. CVDA values are as follows:

IIOP Specifies that the work is being run for an IIOP request.

SOAP Specifies that the work is being run for a Web service request.

If you do not specify a value for this option, the type of work defaults to IIOP.

Conditions

END

RESP2 values:

- 2 There are no more work requests of this type present.

ILLOGIC

RESP2 values:

- 1 You have issued a START command when a browse of work requests is already in progress, or you have issued a NEXT or an END command when a browse of work requests is not in progress.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

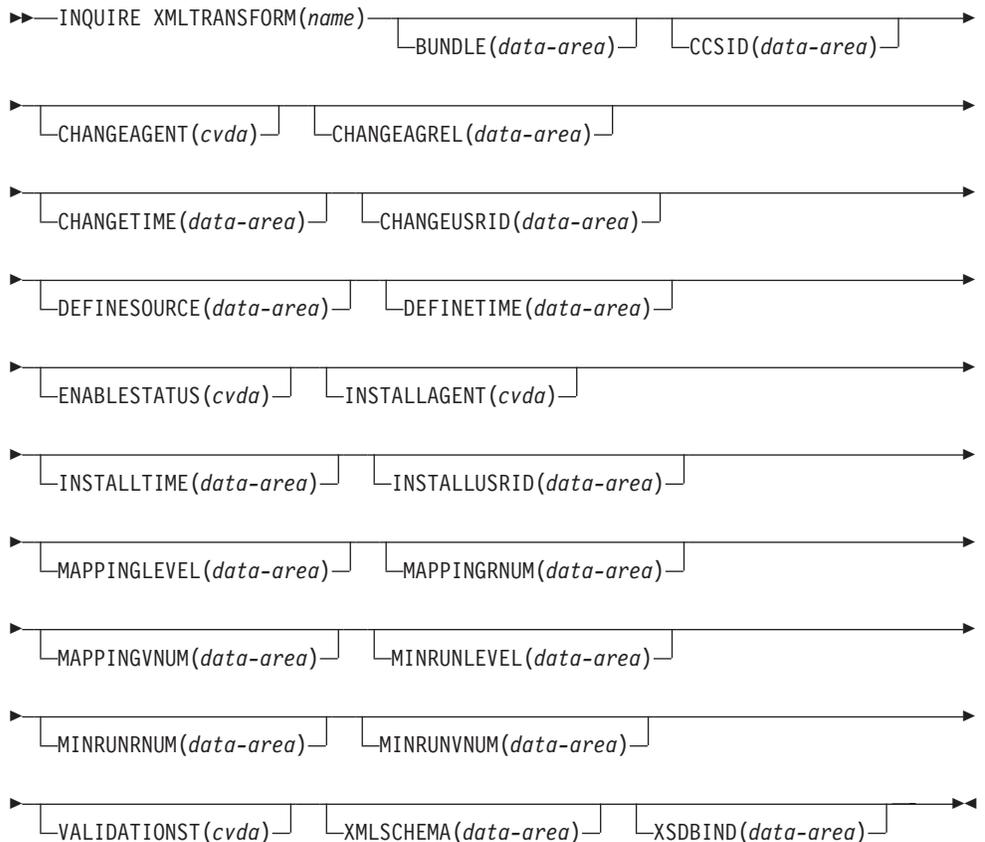
RESP2 values:

- 3 The specified work request is not present in the system.

INQUIRE XMLTRANSFORM

Use the **INQUIRE XMLTRANSFORM** command to retrieve information about an installed XMLTRANSFORM resource.

INQUIRE XMLTRANSFORM



Conditions: NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the **INQUIRE XMLTRANSFORM** command to retrieve information about an installed XMLTRANSFORM resource. This information can include the state of the XMLTRANSFORM resource and details about the conditions under which the XMLTRANSFORM resource was installed, such as which mapping level was used.

Browsing

You can browse through all the XMLTRANSFORM resources installed in your system by using the browse options, START, NEXT, and END, on the **INQUIRE XMLTRANSFORM** command.

The resource signature

You can use this command to retrieve the resource signature fields. You can use these fields to manage resources by capturing details of when the resource was defined, installed, and last changed. For more information, see Auditing resources in the Resource Definition Guide. The resource signature fields are CHANGEAGENT, CHANGEAGREL, CHANGETIME, CHANGEUSRID,

DEFINESOURCE, DEFINETIME, INSTALLAGENT, INSTALLTIME, and INSTALLUSRID. See Summary of the resource signature field values in the Resource Definition Guide for detailed information about the content of the resource signature fields.

Options

BUNDLE(*data-area*)

Returns the 8-character name of the bundle from which the XMLTRANSFORM was installed.

CCSID(*data-area*)

Returns the coded character set identifier (CCSID) that is used to encode the character data in the application data structure at run time. This value is set using the optional CCSID parameter in the XML assistant when the XML binding file is generated. The CCSID is a value of up to 8 characters. If CCSID value is not specified, CICS uses the default CCSID that is specified by the **LOCALCCSID** system initialization parameter.

CHANGEAGENT(*cvda*)

Returns a CVDA value that identifies the agent that made the last change to the resource definition. The possible values are as follows:

CREATESPI

The resource definition was last changed by an **EXEC CICS CREATE** command.

CSDAPI

The resource definition was last changed by a CEDA transaction or the programmable interface to DFHEDAP.

CSDBATCH

The resource definition was last changed by a DFHCSDUP job.

DREPAPI

The resource definition was last changed by a CICSplex SM BAS API command.

DYNAMIC

The resource was defined by an ATOMSERVICE resource.

CHANGEAGREL(*data-area*)

Returns a 4-digit number of the CICS release that was running when the resource definition was last changed.

CHANGETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was last changed. For more information about the format of the ABSTIME value, see **FORMATTIME** in CICS Application Programming.

CHANGEUSRID(*data-area*)

Returns the 8-character user ID that ran the change agent.

DEFINESOURCE(*data-area*)

Returns the 8-character source of the resource definition. The **DEFINESOURCE** value depends on the **CHANGEAGENT** value. For more information, see Summary of the resource signature field values in the Resource Definition Guide.

DEFINETIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource definition was created.

ENABLESTATUS(*cvda*)

Returns a CVDA indicating the state of the XMLTRANSFORM.

DISCARDING

A DISCARD command has been issued for the XMLTRANSFORM.

ENABLING

The XMLTRANSFORM is in the process of being enabled.

ENABLED

The XMLTRANSFORM is enabled and available for use.

DISABLING

The XMLTRANSFORM is in the process of being disabled. It is not available for further use, but inflight activity will be allowed to complete.

DISABLED

The XMLTRANSFORM is disabled and is not available for use.

INSTALLAGENT(*cvda*)

Returns a CVDA value that identifies the agent that installed the resource. The possible values are as follows:

BUNDLE

The resource was installed by a bundle deployment.

DYNAMIC

The resource was installed by an ATOMSERVICE resource.

INSTALLTIME(*data-area*)

Returns an ABSTIME value that represents the time stamp when the resource was installed.

INSTALLUSRID(*data-area*)

Returns the 8-character user ID that installed the resource.

MAPPINGLEVEL(*data-area*)

Returns an 8-byte character string of the mapping level that was used when the XML binding file was produced. The value of the mapping level is 1.0, 1.1, 1.2, 2.0, 2.1, 2.2, or 3.0.

MAPPINGRNUM(*data-area*)

Returns a fullword binary value of the release number for the mapping level that was used when the XML binding file was produced. The value of the release number is 0, 1, or 2.

MAPPINGVNUM(*data-area*)

Returns a fullword binary value of the version number for the mapping level that was used when the XML binding file was produced. The value of the version number is 1, 2, or 3.

MINRUNLEVEL(*data-area*)

Returns an 8-byte character string of the minimum runtime level that is required to install the XMLTRANSFORM in CICS. The value of the runtime level is 3.0.

MINRUNRNUM(*data-area*)

Returns a fullword binary value for the release number for the minimum runtime level that is required to install the XMLTRANSFORM in CICS. The value of the release number is 0.

MINRUNVNUM(*data-area*)

Returns a fullword binary value for the version number for the minimum runtime level that is required to install the XMLTRANSFORM in CICS. The value of the version number is 3.

VALIDATIONST(*cvda*)

Indicates whether full validation is enabled for the XMLTRANSFORM resource. CVDA values are as follows:

VALIDATION

Full validation is enabled.

NOVALIDATION

Full validation is disabled.

Because validating an XML message against its schema incurs considerable processing overhead, typically you will specify VALIDATIONST(NOVALIDATION). If VALIDATIONST(NOVALIDATION) is specified, checking is performed to ensure that the message contains well-formed XML, but with no guarantee that the XML is valid.

Full validation ensures that the XML in the message is valid with respect to the XML schema; you might want to specify VALIDATIONST(VALIDATION) when you are developing an application.

XMLTRANSFORM(*data-value*)

Specifies the 1- to 32-character name of the XMLTRANSFORM about which you are inquiring.

XMLSCHEMA(*data-area*)

Returns the name of the associated XML schema file. The data area is 255 characters long. If the name is shorter than 255 characters, CICS pads the data area with trailing blanks.

XSDBIND(*data-area*)

Returns the name of the XML binding file. The data area is 255 characters long. If the name is shorter than 255 characters, CICS pads the data area with trailing blanks.

Conditions**NOTAUTH**

RESP2 values are:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

NOTFND

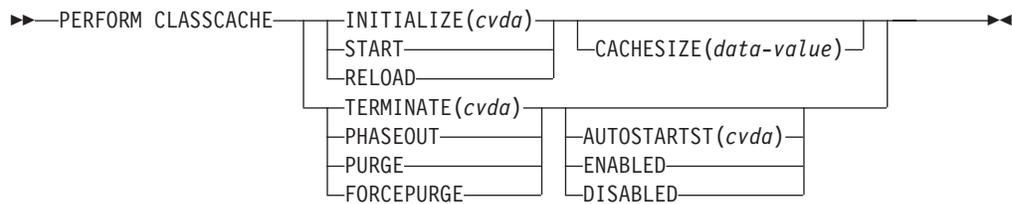
RESP2 values are:

- 3 The XMLTRANSFORM cannot be found.

PERFORM CLASSCACHE

Initialize and terminate the shared class cache that is used by pooled JVMs in the CICS region.

PERFORM CLASSCACHE



Conditions: INVREQ, NOTAUTH

This command is threadsafe.

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

You can use the **PERFORM CLASSCACHE** command to start, reload, and terminate the shared class cache. This class cache is used by pooled JVMs in the CICS region. You cannot use this command for class caches that are used by JVM servers.

While you are performing one of these operations, you can use other options on the command to set attributes of the shared class cache, as follows:

- When you are initializing the shared class cache (using the **START** or **RELOAD** option), you can change its size (**CACHESIZE** option). The new setting overrides the size originally set by the **JVMCCSIZE** system initialization parameter .
- When you are terminating the shared class cache (using the **PHASEOUT**, **PURGE**, or **FORCEPURGE** option), you can set the status of autostart (**AUTOSTARTST** option). If you do not want the shared class cache to start up again until you enter an explicit command, you can use this option to ensure that autostart is disabled. The new setting overrides the autostart status that was originally set by the **JVMCCSTART** system initialization parameter.

When you change the size or autostart status of the shared class cache while CICS is running, subsequent warm or emergency CICS restarts use the changed setting. However, the shared class cache normally persists across warm or emergency restarts, so the changed setting normally has no effect in this situation. If the system is initial or cold started, the settings are always taken from the relevant system initialization parameters.

Options

AUTOSTARTST(*cvda*)

Sets the status of autostart for the shared class cache. The CVDA values are as follows:

ENABLED

When autostart is enabled, if the shared class cache has been stopped or has not yet been started on this CICS run, the shared class cache is started as soon as CICS receives a request to run a Java application in a JVM with a profile that requires the use of the shared class cache.

DISABLED

When autostart is disabled, if the shared class cache has been stopped or has not yet been started on this CICS run, an explicit **PERFORM CLASSCACHE** command is required to start it. If the status of the shared class cache is STOPPED and autostart is disabled, and CICS receives a request to run a Java application in a JVM with a profile that requires the use of the shared class cache, the application cannot run.

CACHESIZE(*data-value*)

Specifies the size in bytes of the shared class cache that is to be started or reloaded as a doubleword binary value. The minimum size that you can specify is 8 MB (8388608 bytes).

INITIALIZE(*cvda*)

Initializes the shared class cache. You can use this option either to start a shared class cache in your CICS region when the shared class cache is stopped (START), or to start a new shared class cache to replace the shared class cache that is active (RELOAD). The CVDA values are as follows:

START

Specifies that the shared class cache is to be started. Use this option to create a new class cache when the status of the shared class cache is STOPPED. If specified, the CACHESIZE is used in the creation of the cache. The shared class cache can only be started after at least one JVM has been created in the CICS region, so that CICS can identify the Java version that is required. CICS retains the Java version across a warm or emergency start. If you issue the **PERFORM CLASSCACHE START** command after a cold start and before any JVMs have been created, CICS validates the parameters but does not start the shared class cache.

RELOAD

Specifies that the shared class cache is to be reloaded. Use this option to create a new class cache when the status of the shared class cache is STARTED (the option only works if the shared class cache has been started).

If specified, the CACHESIZE is used in the creation of the cache. JVMs, both those that are already allocated to tasks and those that are allocated to tasks after you issue the command, continue to use the existing shared class cache until the new shared class cache is ready. When the new shared class cache is ready, subsequent requests for JVMs are given a JVM that uses the new cache. These new JVMs are started when they are requested by applications, and they replace the JVMs that are using the old shared class cache. The JVMs that are using the old shared class cache are allowed to finish running their current Java programs, and then they are terminated. The old shared class cache is deleted when all the JVMs that are using it have been terminated.

TERMINATE(*cvda*)

Specifies that the shared class cache is to be stopped and deleted. All JVMs using the shared class cache are terminated. No more JVMs can use the shared class cache, and it is deleted when all the JVMs that were using it have been terminated. This command operates on JVMs using the most recent shared class cache, and on JVMs using any old shared class caches that are present in the system. JVMs running independently of the shared class cache are not affected by this command.

If autostart is enabled for the shared class cache, a new shared class cache is started as soon as a new JVM requests it. If autostart is disabled, the **PERFORM**

CLASSCACHE INITIALIZE(START) command (or the equivalent CEMT command) must be entered if you want to start a new shared class cache. If you do not restart the shared class cache, requests to run a Java application in a JVM whose profile requires the use of the shared class cache will fail. Specify the **AUTOSTARTST** option to set the status of autostart.

You can repeat the command, using the **PURGE** or **FORCEPURGE** option, to make a stronger attempt to purge the tasks using the JVMs. However, you should only repeat the command if autostart for the shared class cache is disabled. If autostart is enabled, and you repeat the command to terminate the shared class cache, the command could operate on the new shared class cache that has been started by the autostart facility, and terminate it. The CVDA values for the **TERMINATE** option are as follows:

PHASEOUT

All JVMs using the shared class cache are marked for deletion. The JVMs are deleted when they finish running their current Java programs. The shared class cache is deleted when all the JVMs that were using it have been terminated.

PURGE

All tasks using JVMs that are using the shared class cache are terminated by the **SET TASK PURGE** command, and the JVMs are terminated. The shared class cache is deleted when all the JVMs that were using it have been terminated.

FORCEPURGE

All tasks using JVMs that are using the shared class cache are terminated by the **SET TASK FORCEPURGE** command, and the JVMs are terminated. The shared class cache is deleted when all the JVMs that were using it have been terminated.

Conditions

INVREQ

RESP2 values:

- 4 An invalid value was specified for **AUTOSTARTST**.
- 5 **TERMINATE** was specified for a shared class cache that is already stopped.
- 6 **START** was specified for a shared class cache that is not stopped.
- 7 **RELOAD** was specified for a shared class cache that is not started.
- 9 The value specified for **CACHESIZE** is too small.

NOTAUTH

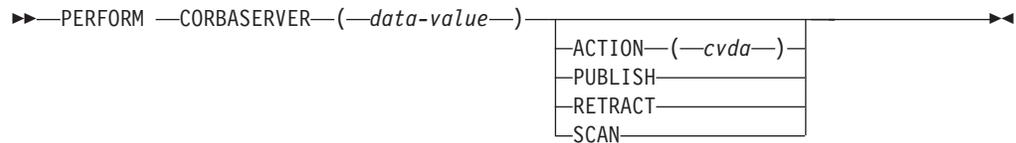
RESP2 value:

- 100 The user associated with the issuing task is not authorized to use this command.

PERFORM CORBASERVER

Perform certain actions against a specified CorbaServer.

PERFORM CORBASERVER



Conditions: DUPRES, INVREQ, NOTAUTH, NOTFND

Description

The PERFORM CORBASERVER command allows you to:

- Publish or retract:
 - All beans installed in the specified CorbaServer
 - The Generic Factory Interoperable Object Reference (GenFacIOR) of the specified Corbaserver

Publishing a bean means binding a reference to the home of the bean in the namespace. The naming context in which the bean is bound is named, relative to the initial context defined for the CICS region, using a concatenation of the JNDI PREFIX attribute of the CorbaServer and the name of the bean.

The Generic factory is bound with the name GenericFactory concatenated to the PREFIX attribute of the Corbaserver.

- Scan the CorbaServer's deployed JAR file directory for new or updated deployed JAR files.

If no action is specified, the command has no effect.

You can issue a PERFORM CORBASERVER command only when the CorbaServer is in a steady state—that is, when it is in ENABLED or DISABLED state, but *not* when it is in ENABLING, DISABLING, or DISCARDING state.

In addition to the traditional DFSMS-managed data sets, such as partitioned data sets (PDS), CICS requires access to z/OS UNIX System Services and its file system in order to launch a JVM. This requires that the CICS region userid is authorized to z/OS UNIX System Services to enable CICS to access the required z/OS UNIX directories listed in the JVM profile and JVM properties file.

Options

CORBASERVER(*data-value*)

specifies the 4-character name of a CorbaServer.

ACTION(*cvda*)

returns a CVDA value indicating the action to be taken on the CorbaServer. CVDA values are:

PUBLISH

publish all the beans installed in the CorbaServer. Publishing a bean means binding a reference to the home of each bean in a namespace. The naming context in which the bean is bound is named, relative to the initial context defined for the CICS region, using a concatenation of the JNDI prefix attribute of the CorbaServer and the name of the bean.

The Generic Factory Interoperable Object Reference (GenFacIOR) of the CorbaServer is also bound in the namespace in the same way.

RETRACT

retract all the beans installed in the CorbaServer. Retracting a bean means unbinding a reference to the home of the bean from the namespace. The naming context in which the bean is bound is named, relative to the initial context defined for the CICS region, using a concatenation of the JNDI prefix attribute of the CorbaServer and the name of the bean.

The GenFacIOR of the CorbaServer is also unbound.

SCAN scan the CorbaServer's deployed JAR file directory (also known as the *pickup directory*) for new or updated deployed JAR files.

If CICS finds any new deployed JAR files in the pickup directory, it copies them to its shelf directory and dynamically creates and installs DJAR definitions for them.

If CICS finds any updated deployed JAR files in the pickup directory, it updates both the LASTMODTIME, DATESTAMP, and TIMESTAMP attributes of the installed DJAR definitions and the shelf copies of the deployed JAR files.

Note:

1. CICS ignores any deployed JAR files in the pickup directory that have the same name *and* the same date and time stamps as currently-installed DJAR resources. A deployed JAR file with the same name but a later date-and-time stamp than an installed DJAR is treated as an update.
2. An error in installing one deployed JAR file has no effect on the success or failure of installing any others in the deployed JAR file directory.

Conditions

DUPRES

RESP2 values:

29 During a scan, one or more deployed JAR files failed to install due to a DJAR naming conflict. This error can occur for two reasons:

1. The name of a DJAR definition created by the scanning mechanism conflicts with that of a previously-installed dynamic definition that targets a different CorbaServer. The scanned-in definition cannot be treated as an update, because the installed definition refers to a different CorbaServer.
2. The name of a DJAR definition created by the scanning mechanism conflicts with that of a statically-installed definition. You cannot use the scanning mechanism to update a static DJAR definition.

To avoid this error:

1. If your CICS region contains multiple CorbaServers, use a different pickup directory for each CorbaServer.
2. Do not place identically-named deployed JAR files into multiple pickup directories in the same CICS region.
3. If you use static DJAR definitions, do not give them names that might conflict with those created by the scanning mechanism.

For more information about DJAR naming conflicts and how to avoid them, see the *CICS Resource Definition Guide*.

INVREQ

RESP2 values:

- 5 The specified CVDA value is invalid; that is, it is neither PUBLISH, RETRACT, nor SCAN.
- 9 The state of the CorbaServer is not ENABLED.
- 10 At least one deployed JAR file belonging to the CorbaServer is in a state other than INSERVICE.
- 13 CorbaServer publish error.
- 14 CorbaServer retract error.
- 25 CorbaServer scan error.
- 26 A scan of this CorbaServer's deployed JAR file directory is already in progress.
- 27 The CorbaServer's deployed JAR file directory (DJARDIR) could not be accessed.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

- 2 The named CorbaServer cannot be found.

PERFORM DELETSHIPED

Delete inactive shipped terminal definitions.

PERFORM DELETSHIPED

▶▶—PERFORM DELETSHIPED—▶▶

Conditions: NOTAUTH

Description

The PERFORM DELETSHIPED command causes immediate invocation of the CICS mechanism for deleting inactive shipped terminal definitions. It does **not** reset the interval at which this mechanism is normally invoked; that is, it does not affect the time remaining until the next automatic invocation.

A shipped definition is inactive if the terminal has not been used locally for a specified period of time and no task is waiting to be attached which requires the terminal. You can determine the length of time a shipped terminal must remain unused to be eligible for deletion and the interval at which CICS checks for such terminals with the INQUIRE DELETSHIPED command, and you can set these values with the SET DELETSHIPED command. For more information about shipped definitions, see the *CICS Intercommunication Guide* and the *CICS Resource Definition Guide*.

Conditions

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

PERFORM DJAR

Perform certain actions against a specified DJAR.

PERFORM DJAR

```
▶▶—PERFORM —DJAR—(—data-value—)————▶▶
|—ACTION—(—cvda—)—|
|—PUBLISH—|
|—RETRACT—|
```

Conditions: INVREQ, NOTAUTH, NOTFND

Description

The PERFORM DJAR command allows you to publish or retract all beans installed in a specified deployed JAR file. If no action is specified, the command has no effect.

In addition to the traditional DFSMS-managed data sets, such as partitioned data sets (PDS), CICS requires access to z/OS UNIX System Services file system in order to launch a JVM. This requires that the CICS region userid is authorized to z/OS UNIX System Services to enable CICS to access the required z/OS UNIX directories listed in the JVM profile and JVM properties file.

Options

DJAR(*data-value*)

specifies the 1–32 character name of a deployed JAR file.

ACTION(*cvda*)

returns a CVDA value indicating the action to be taken on the deployed JAR file. CVDA values are:

PUBLISH

publish all the beans installed from the deployed JAR file. Publishing a bean means binding a reference to the home of the bean in a namespace. The naming context in which the bean is bound is named, relative to the initial context defined for the CICS region, using a concatenation of the JNDI prefix attribute of the deployed JAR file's CorbaServer and the name of the bean.

RETRACT

retract all the beans installed from the deployed JAR file. Retracting a bean means unbinding a reference to the home of the bean from the namespace. The naming context in which the bean is bound is named, relative to the initial context defined for the CICS region, using a concatenation of the JNDI prefix attribute of the deployed JAR file's CorbaServer and the name of the bean.

Conditions

INVREQ

RESP2 values:

- 5 The specified CVDA value is invalid, that is, it is neither PUBLISH nor RETRACT.
- 10 The state of the deployed JAR file is not INSERVICE.
- 16 djar publish error.
- 17 djar retract error.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

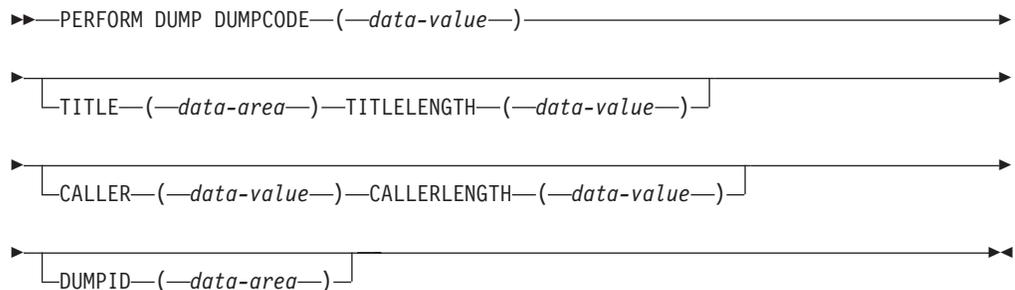
RESP2 values:

- 3 djar not found.

PERFORM DUMP

Request a system dump of CICS.

PERFORM DUMP



Conditions: INVREQ, IOERR, NOSPACE, NOSTG, NOTAUTH, SUPPRESSED, SYSBUSY

Description

The PERFORM DUMP command requests a system dump (an MVS SDUMP) of the CICS region in which it is issued.

The system dump table entry for the dump code specified in the DUMPCODE option determines the processing that occurs on a PERFORM DUMP command: whether a dump is taken, whether the request is propagated for related CICS regions in a sysplex environment, and whether shutdown occurs. If there is no entry for the dump code you specify, CICS creates a temporary one by using default values. See the INQUIRE SYSDUMPCODE command for more information about this process and The dump code options you can specify the *CICS Problem Determination Guide* for general information about the system dump table.

While an MVS SDUMP is being taken, all other CICS activity ceases. The program issuing the command does not regain control until the dump is complete, and then only if the dump does not cause CICS to shut down.

Options

CALLER(*data-value*)

Specifies the text that appears after 'CALLER' in the summary of dump domain information at the top of the dump. This text can be up to 8 characters long. It is intended to identify the source of the request for the dump, but is not restricted to that purpose.

CALLERLENGTH(*data-value*)

Specifies, as a fullword binary value, the number of characters in the CALLER text.

DUMPCODE(*data-value*)

Specifies the 8-character dump code for this dump request, which determines the system dump table entry used in processing it.

The code can be either CICS-defined or user-defined. Most CICS codes are a CICS message identifier with the initial 'DFH' removed, but there are a few additional ones. *CICS Messages and Codes Vol 1* lists all CICS messages and also the additional codes (under "System dump codes").

User-defined codes can be any character string that does not contain leading or imbedded blanks.

CICS provides system dump table entries for some CICS-defined codes and builds them as needed for others. The installation can provide entries for user-defined codes, or CICS builds temporary entries, as previously explained.

DUMPID(*data-area*)

Returns a 6- to 9-character dump identifier generated for this particular dump. The format of the identifier is *xxxx/yyyy*, where *xxxx* represents the **dump run number**, *yyyy* is the **dump count**, and the slash (/) symbol is a separator character. The dump identifier is generated as follows:

Dump run number

A number in the range 1 - 9999. (Leading zeros are not used for this number, which is why the dump ID can vary from 6 to 9 characters.) The dump run number begins at 1 when you first start CICS with a newly-initialized local catalog, and is incremented by 1 each time you restart CICS. The dump run number is saved in the local catalog when you perform a normal shutdown, but is reset if you start CICS with a **START=INITIAL** or **START=COLD** system initialization parameter.

Dump count

A number in the range 0001 through 9999. (Leading zeros are required in the dump ID.) This is the number assigned to the dump in this run of CICS, starting at 0001 for the first dump, and incremented by 1 with each dump taken.

TITLE(*data-area*)

Is the text that is printed as a title in the summary of dump domain information at the top of the dump. It can be up to 80 characters long.

TITLELENGTH(*data-value*)

Specifies, as a fullword binary value, the number of characters in the TITLE text.

Conditions

INVREQ

RESP2 values:

- 6 TITLELENGTH is greater than 80 bytes.
- 7 CALLERLENGTH is greater than 8 bytes.
- 13 The DUMPCODE contains leading or imbedded blanks.

IOERR

RESP2 values:

- 9 CICS is not authorized by MVS to take dumps.
- 10 An error occurred during system dumping.
- 12 MVS cannot process the dump because there is no dump data set or because it is full.
- 13 An error occurred in the CICS routine that issues MVS SDUMP requests.

NOSPACE

RESP2 values:

- 4 The dump is incomplete due to lack of dump data set space.

NOSTG

RESP2 values:

- 5 CICS cannot complete the dump because of insufficient storage.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SUPPRESSED

RESP2 values:

- 1 The dump was not taken because the number of dumps with this dump code exceeds the maximum for the code.
- 2 The dump was not taken because the system dump table entry for this code indicates no system dump.
- 3 The dump was not taken because it was suppressed by a user exit program.
- 8 The dump was not taken because system dumps are suppressed globally.

SYSBUSY

RESP2 values:

- 11 The MVS dump routine is busy. Retry the command.

PERFORM ENDAFFINITY

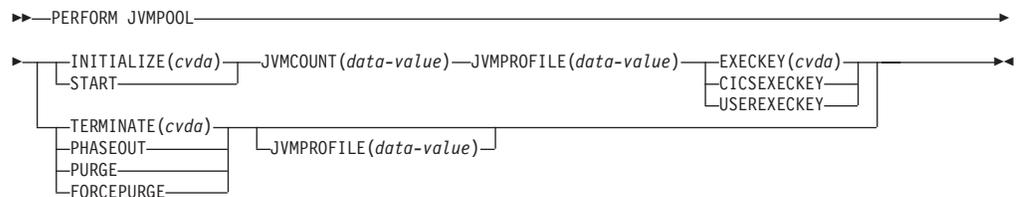
End an affinity owned by CICS.

- 34 NETID was not specified, and cannot be obtained from the installed connection. This may be because the connection does not exist, or because it does not contain a NETID value.
- 35 z/OS Communications Server could not find an affinity for the values input.
- 36 z/OS Communications Server could not end the affinity because the connection had some sessions active.
- 37 See message DFHZC0176. A z/OS Communications Server error prevented the CHANGE ENDAFFIN macro being carried out.

PERFORM JVMPOOL

Start and terminate JVMs in the JVM pool.

PERFORM JVMPOOL



Conditions: INVREQ, NOTAUTH

This command is threadsafe.

Description

You can use the PERFORM JVMPOOL command to start JVMs with your chosen JVM profile and execution key. You can also use the command to terminate all or some of the JVMs in the pool, in order to implement changes to JVM profiles, or to add new application classes.

Options

EXECKEY(cvda)

specifies the execution key for the JVMs. CVDA values are:

CICSSEXECKEY

The JVMs are to execute in CICS key.

USEREXECKEY

The JVMs are to execute in user key. If the system initialization parameter STGPROT=NO is in effect, this is ignored, and the JVMs are started in CICS key.

INITIALIZE(cvda)

specifies a number of JVMs to be started with the JVM profile named by the JVMPROFILE option, and in the execution key named by the EXECKEY option. CICS starts the JVMs asynchronously, so you might not receive notifications of all failures.

The only permitted CVDA value is **START**.

You cannot start up JVMs when the JVM pool's status is set to disabled.

JVMCOUNT (*data-value*)

specifies, as a fullword binary value, the number of JVMs to be started. If the number of JVMs you request, added to the number of JVMs that already exist, would mean exceeding the MAXJVMTCBS limit for the CICS region, CICS returns an error and does not start any JVMs.

JVMPROFILE (*data-value*)

specifies the 8-character name of the JVM profile to be used for the JVMs, padded with trailing spaces if necessary. The name is case-sensitive, and you must enter it using the same combination of upper and lower case characters that is present in the z/OS UNIX file name.

This option is required with the INITIALIZE option. It is optional with the TERMINATE option; if it is not specified, all the JVMs in the pool are terminated.

TERMINATE (*cvda*)

specifies that all or some of the JVMs in the JVM pool are to be terminated. The new JVMs that are started to handle incoming requests incorporate any changes that you made to their JVM profiles, or new application classes that you added. You can start new JVMs manually using the PERFORM JVMPOOL START command, or let CICS start them automatically.

The PERFORM JVMPOOL TERMINATE command does not terminate the shared class cache. If the CICS region has a shared class cache, and you want to terminate it, use the PERFORM CLASSCACHE command to do this. The CVDA values for the TERMINATE option are:

PHASEOUT

JVMs with the selected profile, or all JVMs in the pool, are marked for deletion. The JVMs are deleted when they finish running their current Java program.

PURGE

All tasks using JVMs with the selected profile, or using all JVMs in the pool, are terminated using the SET TASK PURGE mechanism, and the JVMs are terminated.

FORCEPURGE

All tasks using JVMs with the selected profile, or using all JVMs in the pool, are terminated by the SET TASK FORCEPURGE mechanism, and the JVMs are terminated.

Conditions**INVREQ**

RESP2 values:

- 1 TERMINATE is specified with an invalid CVDA value.
- 2 INITIALIZE is specified with an invalid CVDA value.
- 3 A task purge failed on a TERMINATE request.
- 4 EXECKEY is specified with an invalid CVDA value.
- 5 JVMCOUNT is out of range (1-999).
- 6 The number of JVMs specified for JVMCOUNT would cause the total number of JVMs in the CICS region to exceed the MAXJVMTCBS limit.
- 7 INITIALIZE is specified when the status of the JVM pool is DISABLED.

Conditions

DUPRES

RESP2 values:

- 29 During a scan one or more Web service binding file failed to install because of a naming conflict with an existing resource. This error can occur for two reasons:
- The conflict is with an existing definition that is associated with a different PIPELINE. The newly-generated resource cannot be treated as an update, because the existing definition refers to a different PIPELINE.
 - The conflict is with a statically-installed definition. You cannot use the scanning mechanism to update a static Web service binding.

INVREQ

RESP2 values:

- 5 The specified CVDA value is invalid; that is, it is not SCAN.
- 9 The state of the PIPELINE is not ENABLED.
- 10 A scan of this PIPELINE's Web service binding directory is already in progress.
- 11 Read access is denied to the directory specified in the PIPELINE's WSDIR attribute.
- 25 PIPELINE scan error.
- 27 The PIPELINE's Web service binding directory could not be accessed.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

- 3 The named PIPELINE cannot be found.

PERFORM RESETTIME

Reset date and time.

PERFORM RESETTIME

▶▶—PERFORM RESETTIME—▶▶

Conditions: INVREQ, NOTAUTH

Description

The PERFORM RESETTIME command resets the CICS date and time from the MVS system date and time.

Conditions

INVREQ

RESP2 values:

- 1 There is no clock in the system.

NOTAUTH

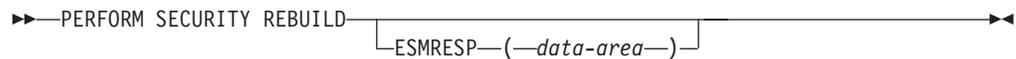
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

PERFORM SECURITY REBUILD

Refresh security information.

PERFORM SECURITY REBUILD



Conditions: INVREQ, IOERR, NORMAL, NOTAUTH, SYSBUSY

Description

The PERFORM SECURITY REBUILD command is a request for CICS security information to be refreshed from its external security manager (ESM) source, so that it reflects any updates made since the information was last retrieved.

Since RACF Version 2.1, the refresh process is automatic; PERFORM SECURITY REBUILD is not required, and has no effect if issued.

If your CICS uses another ESM, the effect of this command depends on the particular ESM.

Options

ESMRESP (*data-area*)

returns a fullword binary field giving the response code from the external security manager. This value is also returned in the RESP2 field of the response code. If an exception condition prevents CICS from invoking the ESM, the ESMRESP value is left unchanged.

If the ESM is not RACF Version 2.1 or later, and the command completes normally (a RESP code value of NORMAL), a RESP2 code of 0 indicates that the profiles have been refreshed. A RESP2 code of 4 means that no action was required, because RACF was at Version 2.1 or later, in which profiles are refreshed automatically.

Conditions

INVREQ

RESP2 values:

- 1 No ESM is installed, or the ESM is inactive.
- 5 The ESM is temporarily inactive and cannot perform the action requested.

IOERR

RESP2 values:

- 3 Error returned from ESM. The return code is in ESMRESP, if the option was used.

NORMAL

RESP2 values:

- 0 Profiles have been refreshed.
- 4 No action is required, because RACF is at Version 2.1 (or later), and so profiles are refreshed automatically.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SYSBUSY

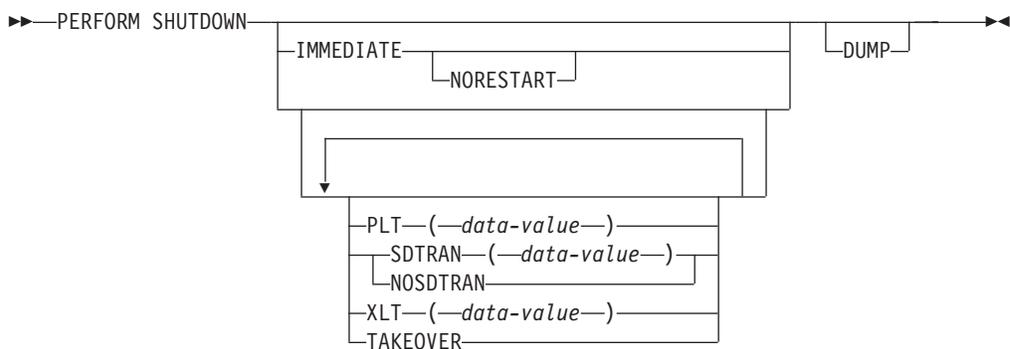
RESP2 values:

- 3 A security rebuild is currently in progress.

PERFORM SHUTDOWN

Shut down the CICS system.

PERFORM SHUTDOWN



Conditions: INVREQ, NOTAUTH, TRANSIDERR

Description

The PERFORM SHUTDOWN command shuts down the CICS system. The shutdown can be either normal (controlled) or immediate. Control does not return to the program issuing the command, unless an exception condition occurs.

In processing this command, CICS invokes the programs in the shutdown program list table (PLT) as part of the task that issued the command. If any program in the list requires a terminal (that is, uses the principal facility), you should not issue the command in a task that does not have one, because the task will abend on the first attempt to use the non-existent terminal. Shutdown will proceed, but the task is backed out to its most recent SYNCPOINT, and the remaining programs in the list will not be executed.

the *CICS Customization Guide* contains more information about PLTs and steps in the shutdown process.

Options

DUMP

specifies that an MVS SDUMP of the CICS region should be taken as part of the shutdown process. In a sysplex environment, dumps of related regions also are taken, if the system dump table entry for the dump code SHUTDOWN, which governs this dump, specifies them.

IMMEDIATE

specifies that CICS is to shut down immediately, terminating all active tasks and z/OS Communications Server sessions abnormally. If IMMEDIATE is **not** specified, CICS shuts down normally, allowing these tasks to complete and quiescing the sessions; it then takes a warm keypoint.

NORESTART

specifies that this CICS region should not be restarted (by MVS automatic restart manager) after the CICS region has completed shutting down. This option applies to immediate shutdowns only.

NOSDTRAN

specifies that no shutdown assist transaction is to be run at CICS shutdown.

PLT(*data-value*)

specifies the 2-character suffix that identifies the PLT for this shutdown. (The table is a load module named DFHPLT followed by this suffix.)

The value "NO" means that no PLT programs are run. If you do not supply a PLT value, the value specified by the PLTSD system initialization parameter, if any, is used. This option applies only to a normal shutdown; the PLT is not run in an immediate shutdown.

SDTRAN(*data-value*)

specifies the 4-character name of the shutdown assist transaction.

The shutdown assist transaction, if specified, is run at CICS warm and immediate shutdown, and can be used to ensure that CICS shuts down in a controlled way, within a reasonable time (by, for example, purging long-running tasks). For details of the default shutdown assist transaction, CESD, see the *CICS Operations and Utilities Guide*.

TAKEOVER

specifies that this CICS system is to be shut down normally, and then the alternate CICS system is to take over. This option is valid only when the system initialization parameter XRF=YES was specified for CICS startup.

XLT(*data-value*)

specifies the 2-character suffix that identifies the transaction list table (XLT) to be used for this shutdown. (The table is a load module named DFHXLT followed by this suffix.)

This table lists the transactions that can be initiated by unsolicited terminal input during the first quiesce stage of a normal shutdown. No other transactions can be initiated from a terminal during shutdown, except for CEMT, CESF, and a small number of other CICS-supplied transactions related to terminals.

This option is meaningful only when IMMEDIATE is not present; no new transactions are accepted during an immediate shutdown. A suffix of "NO" means that no transactions besides those cited above are allowed. If you do not supply an XLT value, the value specified by the XLT system initialization parameter, if any, is used.

Conditions

INVREQ

RESP2 values:

- 1 A normal shutdown was requested when shutdown was already in progress.
- 2 The XLT cannot be found.
- 3 The PLT cannot be found.
- 4 XRF is not in effect.
- 5 The transaction specified on SDTRAN is not enabled for shutdown.
- 6 The transaction specified on SDTRAN is defined as remote.
- 7 The transaction specified on SDTRAN is not enabled.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

TRANSIDERR

RESP2 values:

- 8 The shutdown transaction specified on SDTRAN was not found.

PERFORM STATISTICS RECORD

Record statistics immediately.

PERFORM STATISTICS RECORD



Conditions: IOERR, NOTAUTH, NOTFND

Description

The `PERFORM STATISTICS RECORD` command causes current statistics for the resource types and system functions that you specify to be recorded (written out to the SMF data set). Recording occurs immediately, and is not governed by the system options that control the recording of these statistics at intervals. (See the discussion of **interval statistics** in `SET STATISTICS` in *CICS System Programming Reference*).

Execution of this command does not affect interval or end-of-day statistics either, except when you specify `RESETNOW`, because the counts are not reset unless `RESETNOW` is specified.

You can specify as many types of statistics as you want, or you can request all (the `ALL` option). For each type you request, CICS provides all of the information available (the information that is recorded in interval statistics). For system services, such as dispatch and dynamic transaction backout, CICS keeps summary (global) statistics. For resource types, CICS keeps specific statistics for each installed resource of the type in question, and for some resource types, CICS keeps global counts as well.

The *CICS Performance Guide* contains details about CICS statistics.

Options

ALL

Records statistics for all resource types and system services. This is the same information that is recorded for interval statistics, and includes counts from the user domain, which are not otherwise available with this command.

ATOMSERVICE

Records specific statistics for all `ATOMSERVICE` resources that are installed in the CICS region.

AUTOINSTALL

Records global statistics on the automatic installation of terminal definitions.

BEAN

Records specific statistics for all enterprise beans installed in the CICS region.

BUNDLE

Records specific statistics for all `BUNDLE` resources that are installed in the CICS region.

CAPTURESPEC

Records specific statistics for all capture specifications that are installed in the CICS region.

CONNECTION

Records specific statistics for all `ISC` over `SNA` and `MRO` connections installed in the CICS region.

CORBASERVER

Records specific statistics for all `CorbaServers` installed in the CICS region.

DB2

Records global statistics for the CICS `DB2` connection and specific statistics for each `DB2ENTRY` defined in the CICS region.

DISPATCHER

Records global statistics on the dispatch function, including task counts and concurrency levels and limits.

DOCTEMPLATE

Records specific statistics for each document template installed in the CICS region.

ENQUEUE

Records global statistics for the enqueue manager.

EPADAPTER

Records specific statistics for all EPADAPTER resources that are installed in the CICS region.

EVENTBINDING

Records specific statistics for all EVENTBINDING resources that are installed in the CICS region.

EVENTPROCESS

Records global statistics for event processing.

FEPI

Records global statistics on the front-end programming interface (FEPI) and specific statistics on FEPI connections, targets, and pools.

FILE

Records specific statistics for all files installed in the CICS region.

IPCONN

Records specific statistics for all IPIC connections installed in the CICS region.

JOURNALNAME

Records specific statistics for all journals installed in the CICS region. This parameter replaces the JOURNALNUM parameter. To record specific statistics for all journals installed in the CICS region, you are recommended to use this parameter.

JOURNALNUM

Records specific statistics returned by the JOURNALNAME parameter.

JVMPPOOL

Records global statistics for the JVM pool.

JVMPROFILE

Records specific statistics for all JVM profiles in the CICS region.

JVMPROGRAM

Records specific statistics for all Java programs in the CICS region that run in a JVM.

JVMSERVER

Records specific statistics for all JVMSERVER resources.

LIBRARY

Records specific statistics for all LIBRARY resources.

LSRPOOL

Records specific statistics on all VSAM LSR pools defined in the CICS region, including statistics on the files within the pool additional to the statistics produced by the FILE option.

MONITOR

Records global statistics on the monitor function of CICS.

MQCONN

Records global statistics for the WebSphere MQ connection.

PIPELINE

Records statistics related to installed pipelines.

PROGAUTO

Records global statistics on automatic installation of program definitions.

PROGRAM

Records global and specific statistics for all programs installed in the CICS region, except for Java programs that run in a JVM (for which you can use the JVMPROGRAM option).

PROGRAMDEF

Records the program definition statistics.

RECOVERY

Records global statistics on the recovery manager.

REQUESTMODEL

Records statistics for all REQUESTMODEL definitions installed in the CICS region.

RESETNOW

Resets all statistics to initial values after recording. You can use this option only in conjunction with the ALL option. The definition of the initial value depends on the statistic being kept; see the *CICS Performance Guide* for details.

STATS

Records global statistics about the statistics-gathering function of CICS.

STORAGE

Records global statistics for all CICS dynamic storage subpool areas, and specific statistics by subpool.

STREAMNAME

Records global statistics on the log manager and specific statistics for all the log streams currently connected.

SYSDUMP

Records global statistics on system dumps and specific statistics for each dump code in the system dump code table.

TABLEMGR

Records global statistics on the CICS table manager.

TCLASS

Records specific statistics for every transaction class defined in the CICS region. This option has the same effect as TRANCLASS and is retained for compatibility with older versions of CICS only; use TRANCLASS instead where possible.

TCPIP

Records global statistics on the IP sockets.

TCPIPSERVICE

Records specific statistics for every TCP/IP service installed in the CICS region.

TDQUEUE

Records global statistics for transient data and specific statistics for each queue defined in the CICS region.

TERMINAL

Records specific statistics for each terminal and session installed in the CICS region.

TRANCLASS

Records specific statistics for every transaction class defined in the CICS region.

TRANDUMP

Records global statistics on transaction dumps and specific statistics for each dump code in the transaction dump table.

TRANSACTION

Records global statistics on transactions and specific statistics for each transaction installed in the system.

TSQUEUE

Records global statistics on temporary storage.

URIMAP

Records statistics related to a URIMAP resource.

VTAM (now z/OS Communications Server)

Records global z/OS Communications Server statistics for the CICS region.

WEBSERVICE

Records statistics related to a WEBSERVICE resource.

XMLTRANSFORM

Records statistics related to an XMLTRANSFORM resource.

Conditions**IOERR**

RESP2 values:

- n* Statistics for at least one of the options chosen were not available; usually the reason for this error is corruption of the memory in which they are accumulated. (See note below.)

NOTAUTH

RESP2 values:

- 100** The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

- n* Statistics for at least one of the options chosen were not available because CICS was initialized without support for the function. (See note below.)

Note: When statistics of a requested type are unavailable, CICS raises the IOERR or NOTFND exception condition, as appropriate, but continues through the remaining types, recording as much information as available.

NOTFND is also returned if you attempt to use PERFORM STATISTICS RECORD PROGRAM when you should use PERFORM STATISTICS RECORD JVMPROGRAM. See Collecting statistics for Java programs that run in a JVM for additional information.

The RESP2 value *n* identifies the last type to fail in this way, as follows:

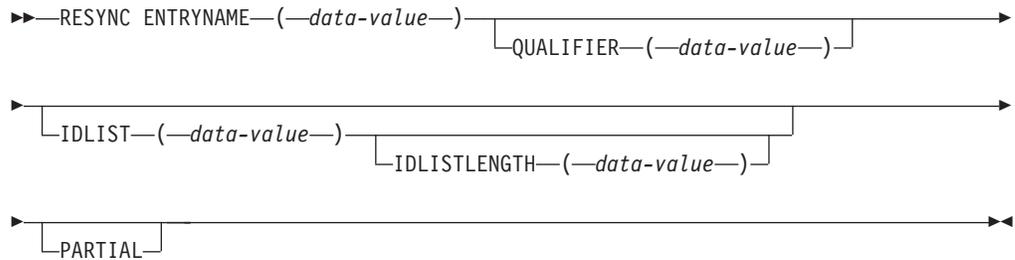
<i>n</i>	Resource type
1	AUTOINSTALL
2	CONNECTION
3	DISPATCHER
6	FILE
8	JOURNALNUM and JOURNALNAME
10	LSRPOOL
11	MONITOR
12	PROGRAM
13	STATS
14	STORAGE
15	SYSDUMP
16	TABLEMGR
18	TCLASS, TRANCLASS
19	TDQUEUE
20	TERMINAL
21	TRANDUMP
22	TRANSACTION
23	TSQUEUE
24	VTAM
25	FEPI
26	PROGAUTO
28	ENQUEUE
29	RECOVERY
30	STREAMNAME
31	DB2
32	TCPIPSERVICE
33	TCPIP
34	REQUESTMODEL
35	CORBASERVER
36	JVMPOOL
37	JVMPROFILE
38	BEAN
39	JVMPROGRAM
40	MQCONN
41	URIMAP

<i>n</i>	Resource type
42	WEBSERVICE
43	PIPELINE
44	DOCTEMPLATE
45	IPCONN
46	LIBRARY
47	PROGRAMDEF
48	BUNDLE
49	ATOMSERVICE
50	EVENTBINDING
51	EVENTPROCESS
52	JVMSERVER
53	XMLTRANSFORM
54	CAPTURESPEC
55	EPADAPTER

RESYNC ENTRYNAME

Determine the disposition of “in doubt” units of work.

RESYNC ENTRYNAME



Conditions: NOTAUTH

This command is threadsafe.

Description

The RESYNC command allows a non-CICS resource manager to determine whether units of work about which it is “in doubt” were committed or backed out.

A resource manager can be in doubt about a unit of work if it has been invoked for the first phase of syncpoint, but not for the second. A failure of either the resource manager or CICS between Phase 1 and Phase 2 leaves the resource manager in doubt about that unit of work.

CICS saves or reconstructs the disposition of any such unit of work until a RESYNC command or an initial start. CICS also saves the disposition of any unit of work about which the resource manager replies “remember” to the second-phase syncpoint invocation, so that if the resource manager cannot commit or roll back as directed, it can request the disposition later for recovery.

To use the saved disposition information, the resource manager must have a record of which units of work are in doubt or “remembered”. It can then issue a RESYNC command with a list of these units of work, either in its task-related user exit program or an associated administrative transaction.

In response, CICS creates a task, CRSY, for each indoubt unit of work in the list. The CRSY task invokes the task-related user exit program once on behalf of its particular unit of work. This invocation is identified to the exit as a phase 2 syncpoint request and as such indicates whether the unit of work was committed or rolled back. The exit program can then relay this information in the form the resource manager requires.

If the resource manager does not want to resynchronize all indoubt units of work at once, it should specify PARTIAL on the RESYNC command. If it does not, CICS discards disposition information for all the indoubt units of work that are not in the supplied list, but are part of the resource manager's resynchronization set.

Note: A resource manager's resynchronization set is initialized when its task-related user exit is first enabled. It is used when the first non-partial RESYNC command is issued. On completion of the non-partial RESYNC, a new resynchronization set is initialized, for use with the next non-partial RESYNC.

A resource manager is identified by the name of its task-related user exit and, optionally, a qualifier to this name. Use of a qualifier allows multiple instances of the same resource manager to resynchronize independently.

Control is returned to the program that issued the RESYNC command as soon as the CRSY tasks have been scheduled. They run asynchronously, in parallel, according to normal CICS dispatch rules. Consequently, the exit should be enabled, started, and initialized to the point where it can process these invocations before the RESYNC command.

If the exit is not available, a CRSY task will save the disposition of its unit of work, but since this occurs later in time, no exceptional condition occurs on the RESYNC. See the *CICS Customization Guide* for full details about resynchronization invocations of task-related user exits.

If CICS fails for some reason, or an immediate shutdown is performed, the “forget flow” log records that are written in response to a committed flow being returned from an external resource manager are lost. This is because anything other than a controlled CICS shutdown does not call the MVS logger to force the log records onto the logstream. This can lead to units of work being rebuilt on a subsequent emergency restart of CICS, if their links to the external resource managers were not seen to be “forgettable” at the time of the restart. In order to allow CICS to discard such units of work, the external resource manager can issue an EXEC CICS RESYNC command to CICS when it reconnects after the CICS system is restarted. Any units of work that are not passed on the command are treated as no longer required by CICS.

Options

ENTRYNAME(*data-value*)

specifies the 8-character name of the task-related user exit for the resource manager. This is the ENTRYNAME value of the ENABLE command that established the exit, or, if ENTRYNAME was omitted, the PROGRAM value.

IDLIST(*data-value*)

specifies the list of units of work to be resynchronized. Each entry in the list is the *address* of the 8-byte identifier of an indoubt unit of work. The end of the list may be indicated by the high-order bit turned on, or IDLISTLENGTH may be used.

Units of work are identified by the UEPURID value passed to the task-related user exit.

Note: IDLIST is optional, but if you omit it, CICS discards all of the saved disposition information for the resource manager, unless you specify PARTIAL. Not specifying a list and specifying PARTIAL is an illogical combination and results in a NO-OP.

IDLISTLENGTH(*data-value*)

specifies a halfword binary value indicating the length (in bytes, counting 4 bytes per indoubt unit of work) of the address-list.

PARTIAL

specifies that CICS is to retain indoubt resolution data for the UOWs (for this resource manager) that are not passed in the indoubt list. PARTIAL indicates that, at this time, the resource manager wants to resynchronize only a subset of the UOWs about which it is in doubt.

If PARTIAL is not specified, CICS discards resolution data for any UOWs not passed in the indoubt list, but which are part of this resource manager's resynchronization set.

Note: A resource manager's resynchronization set is initialized when its task-related user exit is first enabled. It is used when the first non-partial RESYNC command is issued. On completion of the non-partial RESYNC, a new resynchronization set is initialized, for use with the next non-partial RESYNC.

This includes data for UOWs that CICS itself is in doubt about.

A task-related user exit program can issue multiple partial resyncs during the lifetime of a connection with its external resource manager. However, it should issue only *one* full (that is, non-partial) resync during the lifetime of a connection. This is typically done when the connection is first established. Full resyncs imply deletion of UOWs not mentioned in the IDLIST. Only when the external resource manager is not connected to CICS can it be sure that it has a complete list of UOWs to pass to CICS.

QUALIFIER(*data-value*)

specifies an 8-character qualifier to the ENTRYNAME value, which identifies the particular instance of the resource manager to which the RESYNC command applies. The qualifier is optional; it is intended for systems where more than one copy of a resource manager can be in use.

When it is in use, this value is assigned to a unit of work by the task-related user exit at the time the unit of work takes place, via the UEPRMQUA value in the user exit parameter list. If the RESYNC command specifies a qualifier, CICS uses only disposition information saved with the same QUALIFIER and

ENTRYNAME values. Similarly, it discards saved dispositions only if they have the same two values, were not included in the IDLIST, and PARTIAL was not specified.

Conditions

NOTAUTH

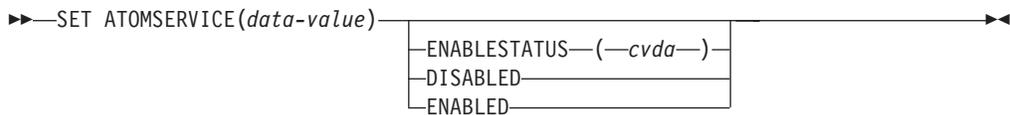
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SET ATOMSERVICE

Enables or disables an ATOMSERVICE definition.

SET ATOMSERVICE



Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Options

ENABLESTATUS(*cvda*)

Sets the ATOMSERVICE definition to enabled or disabled status. CVDA values are:

DISABLED

The ATOMSERVICE definition cannot be accessed by applications. An ATOMSERVICE definition has to be disabled before it can be reinstalled or discarded. If you disable an ATOMSERVICE resource definition, CICS returns an HTTP response to the Web client with a 503 (Service Unavailable) status code.

ENABLED

The ATOMSERVICE definition can be accessed by applications.

Conditions

INVREQ

RESP2 values:

- 9 Invalid ENABLESTATUS value.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

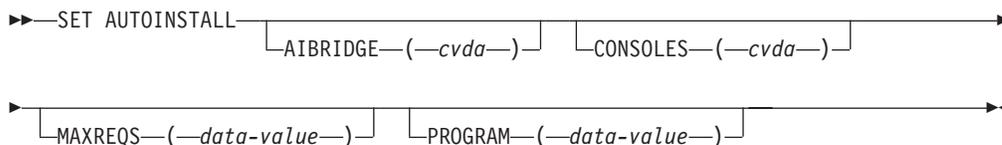
NOTFND

RESP2 values:

SET AUTOINSTALL

Change autoinstall values.

SET AUTOINSTALL



Conditions: INVREQ, NOTAUTH, PGMIDERR

Description

The SET AUTOINSTALL command lets you change some of the values that control the automatic installation (autoinstall) of z/OS Communications Server terminals, APPC sessions, virtual terminals (bridge facilities) used by the 3270 bridge mechanism, and MVS consoles in a CICS region.

Options

AIBRIDGE(*cvda*)

Specifies whether the autoinstall user replaceable program (URM) is to be called for bridge facilities. The CVDA values are:

AUTOTERMID

Bridge facilities are to be defined automatically by CICS. The autoinstall user replaceable program is not to be called.

URMTERMID

The autoinstall user replaceable program is to be called.

CONSOLES(*cvda*)

specifies whether CICS is to autoinstall an MVS console when it receives an MVS MODIFY command from a console that is not defined. The CVDA values are:

PROGAUTO

MVS consoles are to be autoinstalled, and CICS is to call the user autoinstall control program to obtain the termid and other user-specified information.

FULLAUTO

MVS consoles are to be autoinstalled by CICS automatically, without calling the user autoinstall control program. CICS assigns the termid for the console automatically, using the \neg (logical not) symbol as the first character.

NOAUTO

Autoinstall for consoles is not allowed.

MAXREQS(*data-value*)

specifies the largest number of autoinstall requests that can be processed concurrently, as a fullword binary value. The value must be in the range 0-999.

Note: MAXREQS does not limit the total number of terminals that can be installed automatically, only the arrival rate. However, you can prevent automatic installation of any additional terminals by setting MAXREQS to 0. Terminals already autoinstalled are not affected, but if they log off, they cannot log on again while MAXREQS is 0.

PROGRAM(*data-value*)

specifies the 8-character name of the program to be used in the autoinstall process for terminals. You can specify either an installation-specific program or the CICS-supplied default, DFHZATDX.

Note: This program and any programs it invokes must be installed before they can be used in the program autoinstall process. You can do this either with explicit PROGRAM definitions or by autoinstall when some other autoinstall program is in force. Otherwise, the program autoinstall process fails when it is next used, and CICS makes it inactive.

Conditions

INVREQ

RESP2 values:

- 1 z/OS Communications Server is not in use in this system.
- 2 The MAXREQS value is not in the range 0-999.
- 4 One of the modules invoked by DFHZATDX (DFHZATA and DFHZATD) cannot be found.
- 20 CONSOLES has an invalid CVDA value.
- 41 AIBRIDGE has an invalid CVDA value.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

PGMIDERR

RESP2 values:

- 3 The program name cannot be found.

SET BRFACILITY

Release a virtual terminal (bridge facility) used by the 3270 bridge mechanism.

SET BRFACILITY



Conditions: NOTAUTH, NOTFOUND

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The SET BRFACILITY command allows you to request deletion of the virtual terminal (bridge facility) used by the 3270 bridge mechanism.

If a transaction is currently running, the bridge facility will be deleted at the end of the transaction. If the bridge facility is currently AVAILABLE, the facility will be deleted at the next garbage clearance.

When a bridge facility is released, the delete function of the XFAINTU global user exit is driven.

Note: Bridge facilities are deleted only in the region in which the command is issued. Bridge facilities can exist in both router and AOR regions. This command deletes the facility in the region on which it is issued. It does not affect the other region, but this means that the bridge facility can no longer be used. However in order to free up the storage occupied by a bridge facility this command should be issued in both regions. This command can only be issued in the router or AOR region where the bridge facility was created.

Options

BRFACILITY((*data-value*)

specifies the 8-byte facility token of the bridge facility.

TERMSTATUS(*cvda*)

specifies that the bridge facility should be marked for deletion.

RELEASED

The bridge facility is to be deleted.

Conditions

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

NOTFOUND

RESP2 values:

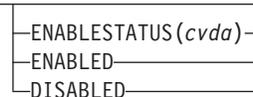
1 The specified bridge facility could not be found.

SET BUNDLE

Change the status of an installed BUNDLE resource.

SET BUNDLE

▶▶—SET BUNDLE(*data-value*)



Conditions: INVREQ, NOTAUTH, NOTFOUND

This command is threadsafe.

Description

Use the **SET BUNDLE** command to change the status of an installed BUNDLE resource. If you enable the BUNDLE resource, CICS tries to enable all of the resources that have been dynamically created by the bundle. CICS cannot enable the BUNDLE resource if one or more of these dynamic resources are in an unusable state.

If you disable the BUNDLE resource, CICS tries to disable all of the resources that have been dynamically created by the bundle.

Options

ENABLESTATUS(*cnda*)

Specifies the status of the BUNDLE:

ENABLED

The BUNDLE and its associated resources are available to CICS.

DISABLED

The BUNDLE and its associated resources are not available to CICS.

Conditions

INVREQ

RESP2 values:

- 6 An attempt was made to enable or disable a BUNDLE resource that is in an invalid state.
- 7 CICS failed to link to the registered bundle callback program.
- 9 BUNDLE cannot be enabled.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

NOTFND

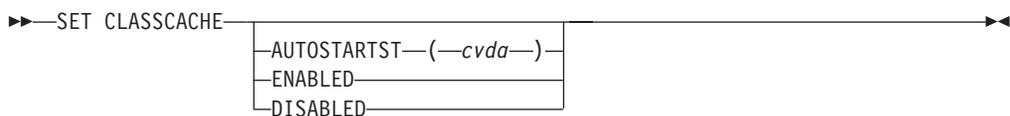
RESP2 values:

- 3 The BUNDLE was not found.

SET CLASSCACHE

Set the status of autostart for the shared class cache that is used by pooled JVMs in the CICS region.

SET CLASSCACHE



Conditions: INVREQ, NOTAUTH

This command is threadsafe.

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

You can use the **SET CLASSCACHE** command to enable and disable the autostart for the class cache that is used by pooled JVMs in the CICS region. You cannot use this command on class caches that are used by JVM servers.

On an initial or cold start of CICS, the status of autostart is taken from the JVMCCSTART system initialization parameter. A value of YES or AUTO gives an autostart status of ENABLED and a value of NO gives an autostart status of DISABLED.

When you change the status of autostart for the shared class cache while CICS is running, subsequent warm or emergency CICS restarts use the changed setting. However, the shared class cache normally persists across warm or emergency restarts, so the changed setting normally has no effect in this situation. If the system is initial or cold started, the status of autostart is always taken from the JVMCCSTART system initialization parameter.

Options

AUTOSTARTST(*cvda*)

specifies the status of autostart for the shared class cache. The CVDA values are as follows:

ENABLED

When autostart is enabled, if the shared class cache has been stopped or has not yet been started on this CICS run, the shared class cache is started as soon as CICS receives a request to run a Java application in a JVM with a profile that requires the use of the shared class cache.

DISABLED

When autostart is disabled, if the shared class cache has been stopped or has not yet been started on this CICS run, an explicit **PERFORM CLASSCACHE** command is required to start it. If the status of the shared class cache is STOPPED and autostart is disabled, and CICS receives a request to run a Java application in a JVM with a profile that requires the use of the shared class cache, the application cannot run.

Conditions

INVREQ

RESP2 values:

4 An invalid value was specified for AUTOSTARTST.

NOTAUTH

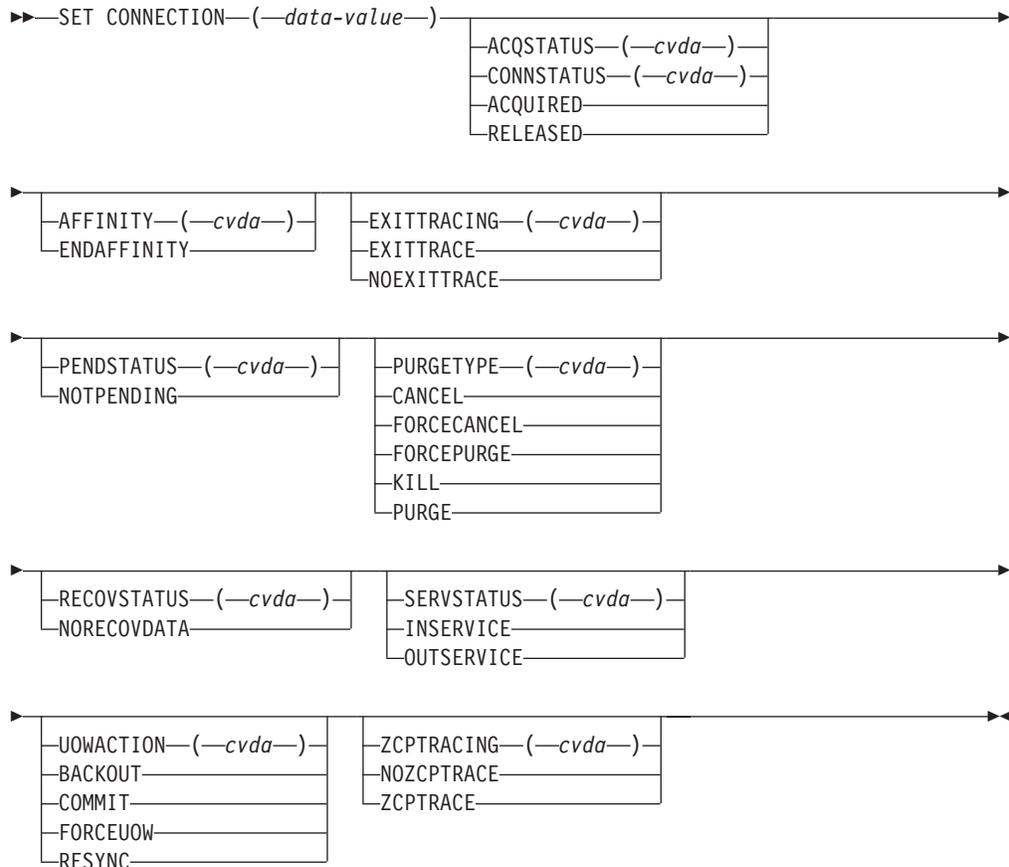
RESP2 value:

100 The user associated with the issuing task is not authorized to use this command.

SET CONNECTION

Change the attributes of an MRO or ISC over SNA connection, or cancel outstanding AIDs.

SET CONNECTION



Conditions: INVREQ, IOERR, NORMAL, NOTAUTH, SYSIDERR

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The SET CONNECTION command allows you to change some of the attributes that define an MRO or ISC over SNA connection. Control returns to the issuing program when the required operation has been scheduled. To get the operation started, it is necessary to relinquish control to CICS.

Note: SET CONNECTION is used to change the attributes of MRO and ISC over SNA connections. See also “SET IPCONN” on page 676. The SET IPCONN command is used to change the attributes of IPIC connections (also known as “IPCONN”).

For information about the different kinds of intercommunication connections, see the *CICS Intercommunication Guide*.

The process of acquiring and releasing the APPC sessions associated with ISC over SNA connections involves starting the LU Services Manager transaction CLS1. In order to pass data to the CLS1 transaction, CICS uses a temporary storage queue with the default prefix DF. If temporary storage queues with the prefix DF are defined as recoverable in your installation, you must follow the SET CONNECTION command by a SYNCPOINT command to end the logical unit of work and allow the SET CONNECTION command to complete.

Options

ACQSTATUS(*cvda*) (APPC only)

This option is retained only for compatibility purposes. Use CONNSTATUS in new applications.

AFFINITY(*cvda*) (APPC and LU61 only)

Specifies, where CICS is a member of a z/OS Communications Server generic resource group, that z/OS Communications Server is to end an affinity owned by CICS. This option is valid only for APPC and LU6.1 connections. The connection must be out of service and, for APPC, in NORECOVDATA state.

The CVDA value is:

ENDAFFINITY

End the affinity.

Notes:

1. There is no facility in z/OS Communications Server for inquiring on affinities, so CICS has no certain knowledge that an affinity exists for a given connection. Whenever there is a possibility that an affinity has been created that must be ended explicitly, CICS issues message DFHZC0177. This message gives the NETNAME and NETID of the suspect connection.
2. If a request to end an affinity is rejected by z/OS Communications Server because no such affinity exists, CICS issues message DFHZC0181.
3. Generic resources and affinities are described in the *CICS Intercommunication Guide*.

CONNECTION(*data-value*)

Specifies, as a 4-character field, the APPC, IRC, or LUTYPE6.1 connection. This is the name of the remote system or region specified in the CONNECTION option of the CEDA DEFINE CONNECTION command.

CONNSTATUS(*cvda*) (APPC only)

Specifies whether to acquire or release sessions with the logical unit represented by the CONNECTION name. To get more detailed information about the availability status of the connection elements, use the INQUIRE MODENAME START, NEXT, and END commands. A connection cannot be both ACQUIRED and OUTSERVICE.

CVDA values are:

ACQUIRED

Sessions are to be acquired.

RELEASED

Sessions are to be released.

For further information about managing APPC connections, see the *CICS Intercommunication Guide*.

Note: CONNSTATUS is applicable to IRC connections for the INQUIRE CONNECTION command but not for the SET CONNECTION command. The CONNSTATUS of an MRO connection is controlled by setting the connection INSERVICE or OUTSERVICE using the SERVSTATUS CVDA.

EXITTRACING(*cvda*) (z/OS Communications Server only)

Specifies whether to trace the activity associated with the terminal exit program for the sessions associated with this connection. CVDA values are:

EXITTRACE

The activity is to be traced.

NOEXITTRACE

The activity is not to be traced.

PENDSTATUS(*cvda*) (APPC and CICS-to-CICS MRO only)

Specifies, for either of the following kinds of connection, that the normal resynchronization process is to be overridden:

- A connection to a CICS Transaction Server for z/OS partner that has performed an initial start
- A connection to a pre-CICS Transaction Server for z/OS partner that has performed a cold start.

The CVDA value is:

NOTPENDING

Forces all indoubt units of work (according to the transaction definition) that were created by the connection before the initial (or cold) start of the partner. It also forgets any resyncs (waitforget UOW-links) that are outstanding for the connection, and created before the initial (or cold) start of the partner.

The PENDING condition indicates the existence of recovery information (either shunted UOWs or decisions remembered for the partner) on a connection that has experienced a lognames mismatch with its partner. For a CICS Transaction Server for z/OS partner, a lognames mismatch indicates that the partner has performed an initial start. For a pre-CICS Transaction Server for z/OS partner, a lognames mismatch indicates that the partner has performed a cold start. In either case, the recovery protocol has been corrupted by a loss of log data at the partner.

It is not possible to set a connection to NOTPENDING state (forcing indoubt and erasing NOFORGET UOWs) until this system has made contact with the partner and received a new logname from it.

Decisions for a whole connection can be forgotten, but that does not affect the memory of a decision for any other connection involved in the UOW.

Note: SET CONNECTION NOTPENDING, SET CONNECTION NORECOVDATA, and SET CONNECTION UOWACTION are mutually exclusive. For advice on which command to use, see the notes following the description of the UOWACTION option.

The exchange lognames function and the resynchronization function are described in the *CICS Intercommunication Guide* and the *Systems Network Architecture—LU6.2 Reference: Peer Protocols* manual.

PURGETYPE(*cvda*)

Specifies how associated transactions are to be purged. CVDA values are:

CANCEL

AIDs queuing for the specified connection are to be canceled.

AIDs representing scheduled and allocated requests waiting in the local CICS system for the specified connection are canceled. However, TD AIDs with an associated triggered task already started are not canceled. In addition, the following CICS system AIDs are not purged unless FORCECANCEL is specified.

Table 6. System AIDs requiring FORCECANCEL to remove them

Remote delete AIDs	
Remote scheduler AIDs	CRSR
LU6.2 service manager 1 AIDs	CLS1
LU6.2 service manager 3 AIDs	CLS3
Remote schedule PURGE AIDs	CRSQ
Resource manager resync AIDs	CRSY
Autoinstall terminal delete AIDs	CATD
Restart terminal delete AIDs	CATR

Message DFHTF0101 is written to CSMT to indicate how many AIDs have been deleted for the connection and how many remain.

When a canceled SCHEDULE request is found to have a precursor in a remote CICS system; that is, the AID was originally scheduled in a remote system, this remote AID is canceled asynchronously.

FORCECANCEL

All AIDs, including system AIDs, queuing for the specified connection are to be canceled. See Table 6 for a list of those system AIDs that require FORCECANCEL to remove them. This can lead to unpredictable results and should be used only in exceptional circumstances.

Note: FORCECANCEL does not remove transient data AIDs with an associated triggered task. These aids may be removed by purging the associated task.

FORCEPURGE

All transactions running on sessions on the connected system are immediately terminated abnormally. This can lead to unpredictable results and should be used only in exceptional circumstances.

In some extreme cases (for example, if an error occurs during backout processing), CICS might terminate abnormally.

For indoubt and shunted UOWs, FORCEPURGE has no effect.

Note: To force shunted UOWs, the operator must issue SET CONNECTION COMMIT, BACKOUT, or FORCE following a FORCEPURGE. This can lead to unpredictable results and should be used only in exceptional circumstances.

KILL The task is to be terminated. System and data integrity is not guaranteed. The KILL option extends the PURGE and FORCEPURGE options. It should be used only after an attempt has been made to PURGE or FORCEPURGE a task. The KILL option does not guarantee integrity of any kind but in some situations it allows you to free up a

stalled region, enabling the region to continue processing. In some cases, for example, if a task is killed during backout processing, CICS terminates abnormally.

PURGE

Transactions running on the connected system are abnormally terminated. Transactions are terminated only if system and data integrity can be maintained. A transaction is not purged if its definition specifies SPURGE=NO, or if the UOW is shunted.

RECOVSTATUS(*cvda*) (APPC only)

Specifies that the normal resynchronization process is to be overridden. The CVDA value is:

NORECOVDATA

Forces all indoubt units of work (according to the transaction definitions), targets any resyncs that were outstanding for the connection, and erases the logname previously received from the partner system. The state of the connection is reset.

Attention: You should use SET CONNECTION NORECOVDATA only in exceptional circumstances. It erases recovery information and may compromise data integrity for units or work that have updated resources on remote systems.

Examples of circumstances in which you might need to use it are:

- You need to discard a connection, or issue a SET CONNECTION ENDAFFINITY command, and it is not possible for the quiesce protocols with the partner system to be completed. (Neither action is possible for an APPC connection if recovery data is outstanding.)
- An operational or logic error results in a logname mismatch for the connection. The connection state must be reset to allow the exchange lognames process to complete.

Note: SET CONNECTION NORECOVDATA, SET CONNECTION NOTPENDING, and SET CONNECTION UOWACTION are mutually exclusive.

SERVSTATUS(*cvda*)

Specifies whether the system is to be placed in service or out of service. CVDA values are:

INSERVICE

The system is to be placed in service; that is, to be available for use.

For an MRO connection, all sessions are placed in service and the following occurs:

- If both the issuing system and the remote system have IRC open, and the remote system has INSERVICE connection definition for the issuing system, the connection is made ACQUIRED (see the note following the description of the CONNSTATUS option).
- Otherwise, the status of the connection is set INSERVICE so that the connection is acquired when the above conditions are met.
- The status of the underlying sessions for a connection is always the same as that for the connection itself.

For an EXCI connection, all receive sessions (or “pipes”) are placed in service and available for use by the client program.

For an ISC APPC connection, the LU Services Manager sessions are placed in service, thereby enabling the connection subsequently to be acquired.

For an ISC LU6.1 connection, all sessions are placed in service.

OUTSERVICE

The connection is to be placed out of service; that is, not available for use.

For a connection, all sessions are placed out of service (immediately if PURGE is specified, or when tasks have terminated if it is not) and the following occurs:

- If an APPC connection is currently ACQUIRED and you specify OUTSERVICE, the command fails with INVREQ and a RESP2 of 2. You must RELEASE the connection before setting OUTSERVICE.
- If any other connection is currently ACQUIRED, the sessions are broken (quiesced). The connection cannot be used until it is once again placed INSERVICE.
- If the connection is currently RELEASED, the status of the connection is set OUTSERVICE and it cannot be used until it is INSERVICE again.
- The status of the underlying sessions for a connection is always the same as that for the connection itself.

For an EXCI connection, all receive sessions (or “pipes”) are placed out of service and are not available for use by the client program.

For an ISC APPC system, this option is valid only if the connection is RELEASED. The LU Services Manager sessions are placed out of service, and the connection cannot be ACQUIRED until it is placed INSERVICE again.

For an ISC LU6.1 connection, all sessions are released and placed out of service: immediately if PURGE or FORCEPURGE is specified; or when tasks have terminated if neither PURGE nor FORCEPURGE is specified. If the response to an INQUIRE CONNECTION command shows OUTSERVICE, it does not imply that the connection has been explicitly set as SET OUTSERVICE; in particular circumstances, you cannot reinstall this connection.

UOWACTION(*cvda*) (APPC parallel-session, CICS-to-CICS MRO, and LU61 only)

Specifies that the normal resynchronization process is to be partially overridden: decisions are taken for any units of work that are indoubt because of a failure of the connection; but the decisions are recorded and any data inconsistencies are reported when the connection is next acquired.

The operation is synchronous with setting the state of the UOW; that is, an INQUIRE UOW following a SET CONNECTION UOWACTION returns the new UOW states. CVDA values are:

BACKOUT

All UOWs shunted because of the failure of this connection are to be backed out.

COMMIT

All UOWs shunted because of the failure of this connection are to be committed.

FORCEUOW

All UOWs shunted because of the failure of this connection are to be forced to BACKOUT or COMMIT, as specified on the ACTION option of the TRANSACTION definition.

RESYNC (MRO-to-CICS Transaction Server for z/OS and later systems, and APPC only)

Any UOWs shunted because of the failure of this connection are to be retried (that is, exchange lognames resynchronization for this connection is to be attempted). This process should normally be started automatically when a connection is acquired or when a UOW is unshunted.

Notes:

1. SET CONNECTION UOWACTION unshunts all units of work that have failed indoubt because of a failure of the connection. Before issuing SET CONNECTION FORCE, you may want to use the SET UOW command to specify commit or backout for each indoubt unit of work explicitly, rather than letting it default. Local procedures will determine the importance of the data and the method of using the INQUIRE UOW, INQUIRE UOWENQ, and INQUIRE UOWLINK commands to establish the correct actions.
2. As far as shunted units of work are concerned, you may use only one of SET CONNECTION UOWACTION, SET CONNECTION NOTPENDING, and SET CONNECTION NORECOVDATA. SET CONNECTION NORECOVDATA should be used only in exceptional circumstances.
3. To force all indoubt units of work caused by a failure of the connection in the same direction, use SET CONNECTION COMMIT or SET CONNECTION BACKOUT.
4. Neither SET CONNECTION UOWACTION nor the SET UOW UOWACTION command clears resync information. If you want to do this, you must use SET CONNECTION NOTPENDING or SET CONNECTION NORECOVDATA.
5. You can issue SET UOW UOWACTION commands *before* issuing SET CONNECTION NOTPENDING or SET CONNECTION NORECOVDATA.

ZCPTRACING(*cvda*) (z/OS Communications Server only)

Specifies whether the z/OS Communications Server control component of CICS is to trace activity on the sessions associated with this connection. CVDA values are:

NOZCPTRACE

z/OS Communications Server ZCP tracing is not to be carried out.

ZCPTRACE

z/OS Communications Server ZCP tracing is to be carried out.

Conditions

INVREQ

RESP2 values:

- 1 ACQSTATUS|CONNSTATUS specified for a non-APPC connection.
- 2 ACQUIRED and OUTSERVICE are specified inconsistently in any of the following ways:
 1. ACQUIRED specified with OUTSERVICE
 2. ACQUIRED specified for OUTSERVICE connection
 3. OUTSERVICE specified for ACQUIRED APPC connection.

- 4. RELEASED and OUTSERVICE specified in the same command for an ACQUIRED connection.
- 3 ACQSTATUS|CONNSTATUS has an invalid CVDA value.
- 4 SERVSTATUS has an invalid CVDA value.
- 5 PENDSTATUS or NOTPENDING was specified for a connection that is not APPC or IRC.
- 6 PURGE was specified for a connection that is not z/OS Communications Server.
- 7 PURGETYPE has an invalid CVDA value.
- 8 PENDSTATUS has an invalid CVDA value.
- 11 SET command named a remote connection.
- 12 EXITTRACING has an invalid CVDA value.
- 13 ZCPTRACING has an invalid CVDA value.
- 14 EXITTRACING|ZCPTRACING specified for a non-z/OS Communications Server connection or z/OS Communications Server not initialized.
- 16 The resource whose name was specified by CONNECTION(*data-value*) is an indirect link.
- 17 ACQSTATUS|CONNSTATUS cannot be set when system initialized with ISC=NO.
- 18 NOTPENDING cannot be set for a connection which has successfully completed Exchange Lognames processing.
- 19 CONNSTATUS cannot be set to ACQUIRED when in the FREEING state.
- 20 COMMIT, BACKOUT, FORCE, or RESYNC is not valid for this type of connection.
- 21 BACKOUT or FORCE was specified, but was unsuccessful. Some UOWs remain shunted for this connection.
- 22 Other SET parameters were included with the CANCEL or FORCECANCEL option.
- 23 The resource whose name was specified by CONNECTION(name) is the local TCT system entry (TCTSE).
- 25 Connection is still in service.
- 26 RECOVSTATUS does not have a value of NORECOVDATA.
- 30 Wrong connection type for ENDAFFINITY. Affinities can exist only on LU6.1 and LU6.2 connections.
- 31 The NETID could not be obtained from the installed connection. Therefore, to end an affinity you must use the PERFORM ENDAFFINITY command.
- 32 See message DFHZC0178. z/OS Communications Server could not end the affinity for a reason other than 35 (NOTFOUND) or 36 (SESSIONS ACTIVE).
- 35 z/OS Communications Server could not find an affinity for this connection.
- 36 z/OS Communications Server could not end the affinity because the connection had some sessions active.
- 37 See message DFHZC0176. A z/OS Communications Server error prevented the CHANGE ENDAFFIN macro being carried out.
- 44 GRSTATUS is not set to REGISTERED or DEREGISTERED. (No generic resource name.)
- 45 NORECOVDATA cannot be set for a connection that is in service.
- 46 NORECOVDATA was specified for a non-APPC connection.

IOERR

RESP2 values:

- 10 Unexpected error.

Changing the setting of AUTOPUBLISH affects only DJAR definitions installed after the SET CORBASERVER command is executed. It has no effect on previously-installed DJAR definitions.

CORBASERVER(*data-value*)

specifies the 4-character name of a CorbaServer.

ENABLESTATUS(*cvda*)

specifies whether to enable or disable the CorbaServer. CVDA values are:

DISABLED

The CorbaServer is to be disabled.

In a CICS region containing multiple CorbaServers, putting one of the CorbaServers into disabled state allows you to apply maintenance to it without bringing down CICS; the other CorbaServers remain available.

Similarly, in a logical EJB/CORBA server containing multiple AORs (each containing multiple CorbaServers), putting a CorbaServer on one of the AORs into disabled state allows you to apply maintenance to the CorbaServer without needing to remove the AOR from the routing set.

You can disable a CorbaServer that is in any state except ENABLING or DISCARDING. When you do so, any currently-executing methods are allowed to continue to completion. Until all its objects have been destroyed or passivated, the CorbaServer enters DISABLING state.

Note that:

- Any instance of an enterprise bean or CORBA stateless object that is not participating in an OTS transaction is destroyed or passivated at the end of the currently-executing method.
- Any instance of an enterprise bean or CORBA stateless object that *is* participating in an OTS transaction is not destroyed or passivated until the end of the OTS transaction; typically, any future method calls against this instance (within the scope of the OTS transaction) will succeed. At the end of the OTS transaction, the instance is destroyed or passivated.

If the workload manager directs new work to an AOR in which the target CorbaServer is in DISABLING or DISABLED state, it is informed that the CorbaServer is disabled.

Note: By “new work” we mean:

- Requests for methods that will run under a new OTS transaction. (By a “new” OTS transaction we mean an OTS transaction in which none of the AORs in the logical server is already participating, before the method call; *not* necessarily an OTS transaction that is started immediately before or after the method call.)
- Requests for methods that will run under no OTS transaction.

ENABLED

The CorbaServer is to be enabled. You can enable a CorbaServer that is in any state except DISABLING or DISCARDING.

SESSBEANTIME(*data-area*)

Fullword area containing the elapsed time period in minutes, in the range 0 through 143999 (99 days, 23 hours, 59 minutes), of inactivity after which a session bean may be discarded. A value of 0 indicates that beans are not timed out.

The new timeout value applies only to session beans created after the SESSBEANTIME value has been set. It does not affect existing session beans.

Conditions

INVREQ

RESP2 values

- 4 The specified session bean timeout value is invalid.
- 15 An attempt was made to enable or disable a CorbaServer that is in an invalid state.
- 21 Object store error.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

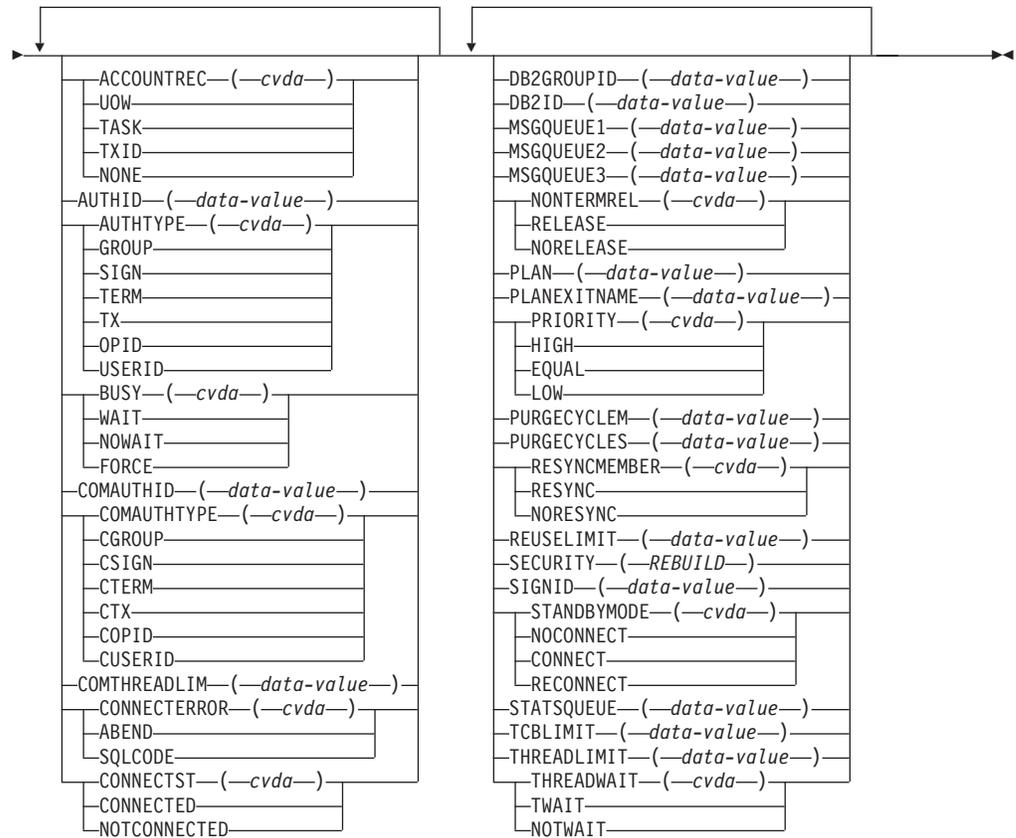
- 2 The specified CorbaServer was not found.

SET DB2CONN

Change information about the attributes of the CICS DB2 Connection.

SET DB2CONN

▶▶—SET DB2CONN—————▶▶



Conditions: NORMAL, NOTAUTH, NOTFND, INVREQ

This command is threadsafe.

For more information about the use of CVDAs, see CICS-value data areas (CVDA) in CICS System Programming Reference.

Description

The SET DB2CONN command also specifies the attributes of the pool and command threads.

Restriction: This command cannot be used in a remote program that is linked by a distributed program link command.

Options

ACCOUNTREC(*cvda*)

Specifies the minimum amount of DB2 accounting required for transactions using pool threads. The specified minimum can be exceeded as described in the following options. CVDA values are:

NONE

No accounting records are required for transactions using pool threads.

DB2 produces at least one accounting record for each thread when the thread is terminated. Authorization changes additionally cause accounting records to be produced.

TXID The CICS DB2 attachment facility causes an accounting record to be produced when the transaction ID that is using the thread changes.

Because pool threads are typically used by a number of different transaction IDs, there is an increased chance that a transaction containing multiple units of work (UOWs) uses a different thread for each UOW (assuming that the thread is released at sync point). In this case an accounting record can be produced per UOW.

TASK The CICS DB2 attachment facility causes a minimum of one accounting record for each CICS task to be produced.

A transaction containing multiple UOWs (assuming that the thread is released at sync point) can use a different thread for each of its UOWs. The result can be an accounting record produced for each UOW.

UOW The CICS DB2 attachment facility causes an accounting record to be produced for each UOW, assuming that the thread is released at the end of the UOW.

AUTHID(*data-value*)

Specifies, as an 8-character name, what ID should be used for security checking for pool threads. If AUTHID is specified, AUTHTYPE cannot be specified.

AUTHTYPE(*cvda*)

Specifies the type of ID that can be used for pool threads. If AUTHTYPE is specified AUTHID cannot be specified. CVDA values are:

GROUP

Specifies the 8-character user ID and the connected group name as the authorization ID. The following table shows how these two values are interpreted by DB2.

IDs passed to DB2	How DB2 interprets values
CICS sign-on user ID (USERID)	Represents the primary DB2 authorization ID.
RACF-connected group name	If the RACF list of group options is not active, DB2 uses the connected group name supplied by the CICS attachment facility as the secondary DB2 authorization ID. If the RACF list of group options is active, DB2 ignores the connected group name supplied by the CICS attachment facility, but the value appears in the DB2 list of secondary DB2 authorization IDs.

To use the GROUP option the CICS system must have SEC=YES specified in the CICS system initialization table (SIT).

If no RACF group ID is available for this USERID, an 8-character field of blanks is passed to DB2 as the group ID.

SIGN Specifies the SIGNID parameter of the DB2CONN should be used as the resource authorization ID.

TERM Specifies the terminal identification (four characters padded to eight) as an authorization ID. An authorization ID cannot be obtained in this manner if a terminal is not connected with the transaction.

If a transaction is started (using a CICS command) and has no terminal associated with it, AUTHTYPE(TERM) should not be used.

TX Specifies the transaction identification (four characters padded to eight) as the authorization ID.

OPID The operator identification associated with the userid that is associated with the CICS transaction is used as the authorization ID (three characters padded to eight).

USERID

The 8-character user ID associated with the CICS transaction is used as the authorization ID.

When the DB2 sample sign-on exit DSN3@.SGN is used with AUTHTYPE(USERID), the exit sends the user ID to DB2 as the primary authorization ID and the RACF group ID to DB2 as the secondary ID. When the sample sign-on exit is used, there is no difference between AUTHTYPE(USERID) and AUTHTYPE(GROUP).

BUSY(*cvda*)

This parameter is valid only with CONNECTST when setting the CICS DB2 connection NOTCONNECTED. CVDA values are:

FORCE

This is like issuing DSNB STOP FORCE; that is, any CICS transactions currently using DB2 are abnormally terminated, and the CICS DB2 attachment facility is stopped. FORCE is mutually exclusive to WAIT and NOWAIT.

NOWAIT

Makes the request asynchronous in nature. Control is returned before the request is complete. NOWAIT is mutually exclusive to WAIT and FORCE.

WAIT

The request is synchronous in nature. Control is only returned when the request is complete. WAIT is mutually exclusive to NOWAIT and FORCE.

Note that a SET DB2CONN NOTCONNECTED WAIT|NOWAIT is a quiesce stop of the CICS DB2 interface. The quiesce waits for existing transactions to finish before stopping the interface.

COMAUTHID(*data-value*)

Specifies, as an 8-character name, which ID should be used for security checking when using command threads. If COMAUTHID is specified, COMAUTHTYPE cannot be specified.

COMAUTHTYPE(*cvda*)

Specifies the type of ID that can be used for security checking when using command threads. If COMAUTHTYPE is specified, COMAUTHID cannot be specified. CVDA values are:

CGROUP

Specifies the 8-character user ID and the connected group name as the authorization ID. The following table shows how these two values are interpreted by DB2.

IDs passed to DB2	How DB2 interprets values
CICS sign-on user ID (USERID)	Represents the primary DB2 authorization ID.
RACF-connected group name	If the RACF list of group options is not active, DB2 uses the connected group name supplied by the CICS attachment facility as the secondary DB2 authorization ID. If the RACF list of group options is active, DB2 ignores the connected group name supplied by the CICS attachment facility, but the value appears in the DB2 list of secondary DB2 authorization IDs.

To use the CGROUP option the CICS system must have SEC=YES specified in the CICS system initialization table (SIT).

If no RACF group ID is available for this USERID, an 8-character field of blanks is passed to DB2 as the group ID.

CSIGN

Specifies the SIGNID parameter of the DB2CONN should be used as the resource authorization ID.

CTERM

Specifies the terminal identification (four characters padded to eight) as an authorization ID. An authorization ID cannot be obtained in this manner if a terminal is not connected with the transaction.

If a transaction is started (using a CICS command) and has no terminal associated with it, the COMAUGHTYPE(TERM) should not be used.

CTX Specifies the transaction identification (four characters padded to eight) as the authorization ID.

COPID

The operator identification associated with the user ID that is associated with the CICS transaction sign-on facility is used as the authorization ID (three characters padded to eight).

CUSERID

The 8-character user ID associated with the CICS transaction is used as the authorization ID.

When the DB2 sample sign-on exit DSN3@.SGN is used with AUTHTYPE(USERID), the exit sends the USERID to DB2 as the primary authorization ID and the RACF group ID to DB2 as the secondary ID. When the sample sign-on exit is used, there is no difference between COMAUGHTYPE(CUSERID) and COMAUGHTYPE(CGROUP).

COMTHREADLIM(*data-value*)

Specifies, as a fullword binary value, the current maximum number of command threads that the CICS DB2 attachment allows to be active before requests overflow to the pool.

CONNECTERROR(*cvda*)

If CICS is not connected to DB2 because the adaptor is in standby mode, describes how this is reported back to an application that has issued a SQL request. CVDA values are:

ABEND

The application is stopped with abend code AEY9.

SQLCODE

The application receives a -923 SQLCODE.

CONNECTST(*cvda*)

Sets the status of the CICS DB2 connection; that is whether, to start or stop the CICS DB2 connection. CVDA values are:

CONNECTED

This is equivalent to issuing DSNB STRT to start the CICS DB2 attachment. If the requested DB2 subsystem is active, control returns when CICS and DB2 have been connected. If the requested DB2 subsystem is not active, the response returned is dependent on the

setting of STANDBYMODE: If DB2 is not initialized, and STANDBYMODE(NOCONNECT) is specified on the DB2CONN, INVREQ, and RESP2=39 is returned. If you specify STANDBYMODE(CONNECT) or STANDBYMODE(RECONNECT), NORMAL with RESP2=38 is returned indicating that the CICS DB2 attachment is in standby mode and connects to DB2 as soon as it becomes active.

NOTCONNECTED

NOTCONNECTED with NOWAIT means initiate quiesce stop of the connection, but return control immediately. NOTCONNECTED WAIT means that control does not return to the application until the CICS DB2 attachment has been stopped. NOTCONNECTED FORCE force stops the connection by force purging transactions currently using DB2. Control is not returned until the connection is stopped.

DB2GROUPID (*data-value*)

Specifies the 4-character name of a data sharing group of DB2 subsystems. CICS attempts to connect to any active member of this group, using group attach. With DB2 Version 10, the 4-character name can be a subgroup name identifying a subset of the data sharing group.

DB2GROUPID can only be changed when CICS is not connected to a DB2 system. Specifying a DB2GROUPID causes the DB2ID in the installed DB2CONN definition to be blanked out. If an individual subsystem's DB2ID is specified in a CEMT or EXEC CICS SET DB2CONN command, or in a DSNC STRT command, this overrides any DB2GROUPID that is set in the installed DB2CONN definition. The DB2GROUPID is blanked out, and must be set again (using CEDA or a SET DB2CONN command) to use group attach. Also note that you cannot set a DB2GROUPID and a DB2ID in the same command — this causes the command to fail.

DB2ID (*data-value*)

Specifies the 4-character name of the DB2 subsystem that the CICS DB2 attachment should connect to. DB2ID can only be changed when CICS is not connected to a DB2 system. Specifying a DB2ID causes the DB2GROUPID in the installed DB2CONN definition to be blanked out, and the DB2GROUPID must be set again to use group attach. If a DB2GROUPID is specified in a CEMT or EXEC CICS SET DB2CONN command, this overrides any DB2ID that is set in the installed DB2CONN definition, and the DB2ID is blanked out. Also note that you cannot set a DB2ID and a DB2GROUPID in the same command — this causes the command to fail.

MSGQUEUE1 (*data-value*)

Specifies, as a 4-character name, the first transient data destination to which unsolicited messages from the CICS DB2 attachment are sent.

MSGQUEUE2 (*data-value*)

Specifies, as a 4-character name, the second transient data destination to which unsolicited messages from the CICS DB2 attachment are sent.

MSGQUEUE3 (*data-value*)

Specifies, as a 4-character name, the third transient data destination to which unsolicited messages from the CICS DB2 attachment are sent.

NONTERMREL (*cvda*)

Specifies whether non-terminal transactions release threads for reuse at intermediate sync points. CVDA values are:

RELEASE

Non-terminal transactions release threads for reuse at intermediate sync points.

NORELEASE

Non-terminal transactions do not release threads for reuse at intermediate sync points.

PLAN(*data-value*)

Specifies the 8-character name of the plan to be used for all threads in the pool. If PLAN is specified, PLANEXITNAME cannot be specified.

PLANEXITNAME(*data-value*)

Specifies the 8-character name of the dynamic plan exit to be used for pool threads. If you change the PLAN and PLANExitname while there are active transactions for the pool, the next time the transaction releases the thread, the plan/exit is determined using the new rules. If PLANExitname is specified, PLAN cannot be specified.

PRIORITY(*cvda*)

Specifies the priority of the pool thread TCBs relative to the CICS main TCB (QR TCB). The thread TCBs are CICS open L8 TCBs. CVDA values are:

HIGH Thread TCBs have a higher priority than the CICS QR TCB.

EQUAL

Thread TCBs have equal priority with the CICS QR TCB.

LOW Thread TCBs have a lower priority than the CICS QR TCB.

PURGECYCLEM(*data-value*)

Specifies, as a fullword binary value, the number of minutes (in the range 00 - 59) in the protected thread purge cycle time. Use this parameter in conjunction with PURGECYCLES.

PURGECYCLES(*data-value*)

Specifies, as a fullword binary value, the number of seconds (in the range 00 - 59) in the protected thread purge cycle time. Use this parameter in conjunction with PURGECYCLEM. The minimum protected thread purge cycle time is 5 seconds and the default is 30 seconds.

A protected thread is not terminated immediately when it is released. It is terminated only after two completed purge cycles, if it has not been reused in the meantime. For example, if the protected thread purge cycle is set to 30 seconds, a protected thread is purged 30 - 60 seconds after that thread is released. An unprotected thread is terminated when it is released (at sync point or end of task) if there are no other transactions waiting for a thread on that DB2ENTRY.

RESYNCMEMBER(*cvda*)

This applies only if you are using group attach, and specifies the strategy that CICS adopts if outstanding units of work are being held for the last DB2 data sharing group member to which CICS was connected. (Units of work which are shunted indoubt are not included in this process, because CICS itself is unable to resolve those units of work at this time. Resynchronization for those UOWs occurs when CICS has resynchronized with its remote coordinator.) CVDA values are:

RESYNC

CICS connects to the same DB2 data sharing group member to resynchronize the outstanding units of work.

NORESYNC

CICS makes one attempt to connect to the same DB2 data sharing group member, and if that attempt fails, CICS connects to any member of the DB2 data sharing group and issues a warning about the outstanding units of work.

REUSELIMIT (*data-value*)

Specifies, as a fullword binary value, a value in the range 0 - 10000 representing the maximum number of times a thread can be reused before it is terminated. The default is 1000. A value of 0 means that there is no limit on the number of times that a thread can be reused.

The reuse limit applies to unprotected threads both in the pool and on a DB2ENTRY, and to protected DB2ENTRY threads.

SECURITY(REBUILD)

Specifies that the CICS DB2 attachment should force all existing threads to signon again at the next thread reuse. It should be used when RACF profiles have been updated by issuing the following commands:

- CEMT PERFORM SECURITY REBUILD for RACF 1.9.2 or earlier
- TSO SETROP TS RACLIST(xxxxxxxx) REFRESH for RACF 2.1 or later

SIGNID (*data-value*)

Specifies the 8-character authorization ID to be used by the CICS DB2 attachment when signing on to DB2 for pool and DB2ENTRY threads specifying AUTHTYPE(SIGN), and command threads specifying COMAUTHTYPE(CSIGN).

STANDBYMODE (*cvda*)

Specifies the action to be taken by the CICS DB2 attachment if DB2 is not active when an attempt is made to start the connection from CICS to DB2. CVDA values are:

NOCONNECT

The CICS DB2 attachment should terminate.

CONNECT

The CICS DB2 attachment goes into standby mode to wait for DB2.

RECONNECT

The CICS DB2 attachment goes into standby mode and waits for DB2. Having connected to DB2, if DB2 later fails the CICS DB2 attachment reverts again to standby mode and then reconnects to DB2 when it comes up again.

STATSQUEUE (*data-value*)

Specifies the 4-character transient data destination for CICS DB2 attachment statistics produced when the CICS DB2 attachment is shut down.

TCBLIMIT (*data-value*)

Specifies, as a fullword binary value, the maximum number of TCBs that can be used to process DB2 requests. CICS creates open TCBs (up to the limit specified in the system initialization parameter MAXOPENTCBS). The TCBLIMIT attribute of the DB2CONN definition governs how many of the open TCBs can be used to access DB2 — that is, how many of them can identify to DB2 and create a connection into DB2.

THREADLIMIT (*data-value*)

Specifies, as a fullword binary value, the current maximum number of pool threads the CICS DB2 attachment allows active before requests are made to wait or are rejected according to the THREADWAIT parameter.

THREADWAIT (*cvda*)

Specifies whether transactions should wait for a pool thread or be abended if the number of active pool threads reach the THREADLIMIT number.

The CICS DB2 attachment issues a unique abend code AD3T, and message DFHDB2011 when THREADWAIT=NO is coded and the number of pool threads is exceeded. CVDA values are:

TWAIT

If all threads are busy, a transaction must wait until one becomes available. A transaction can wait as long as CICS allows it to wait, generally until a thread becomes available.

NOTWAIT

If all threads are busy the transaction is terminated with an abend code AD3T.

Notes:

1. When you change the value of AUTHID, AUTHTYPE, COMAUTHID, COMAUTHTYPE, or SIGNID, a surrogate user security check is invoked if security is active. This ensures that the user ID under which SET is being executed is authorized to act on behalf of the user ID being set.
2. When the SET DB2CONN command is specified all parameters except DB2ID (the connected subsystem) and DB2GROUPID (the group of data sharing DB2 subsystems of which the connected subsystem is a member) can be set when the CICS DB2 attachment is active. DB2ID and DB2GROUPID can only be changed by stopping and restarting the attachment.
3. If you change the PLAN and PLANEXITNAME while there are active transactions for that entry, or the pool, the next time the transaction releases the thread, the plan, or exit is determined using the new rules.

Conditions**NORMAL**

RESP2 values:

- | | |
|-----------|---|
| 38 | Waiting for DB2 (this can occur following a CONNECTST with a CVDA of CONNECT) |
| 55 | Temporarily connected to a DB2 restart-light member for resynchronization purposes only |

NOTAUTH

The user associated with the issuing task is not authorized to use this command, or is not authorized to access this resource.

RESP2 values:

- | | |
|------------|---------------------------------|
| 100 | Command authorization failure |
| 102 | Surrogate authorization failure |
| 103 | Authtype authorization failure |

NOTFND

RESP2 values:

- | | |
|----------|--|
| 1 | There is no DB2CONN currently installed. |
|----------|--|

INVREQ

RESP2 values:

- | | |
|----------|--------------------------|
| 1 | Invalid ACCOUNTREC value |
|----------|--------------------------|

- 2 Invalid AUTHTYPE value
- 3 Invalid BUSY value
- 4 Invalid COMAUTHTYPE value
- 5 Invalid CONNECTERROR value
- 6 Invalid CONNECTST value
- 7 Invalid NONTERMREL value
- 9 Invalid PRIORITY value
- 10 Invalid SECURITY value
- 11 Invalid STANDBYMODE value
- 12 Invalid THREADWAIT value
- 13 Bad characters in AUTHID
- 14 Bad characters in COMAUTHID
- 15 Bad characters in DB2ID
- 16 Bad characters in MSGQUEUE1
- 17 Bad characters in MSGQUEUE2
- 18 Bad characters in MSGQUEUE3
- 19 Bad characters in PLAN
- 20 Bad characters in PLANEXITNAME
- 21 Bad characters in SIGNID
- 22 Bad characters in STATSQUEUE
- 23 Both AUTHID and AUTHTYPE specified
- 24 Both COMAUTHID and COMAUTHTYPE specified
- 25 STANDBYMODE(NOCONNECT) and CONNECTERROR(SQLCODE) specified or CONNECTERROR(SQLCODE) specified when STANDBYMODE is NOCONNECT.
- 26 Both PLAN and PLANEXITNAME specified.
- 27 Invalid ACCOUNTREC value
- 28 COMTHREADLIM exceeds TCBLIMIT or COMTHREADLIM > 2000 or COMTHREADLIM < 0
- 29 Purge cycle too low; that is, < 30 seconds
 - or Purge cycle minutes < 0
 - or Purge cycle seconds < 0
 - or Purge cycle minutes > 59
 - or Purge cycle seconds > 59
- 32 Tcblimit > 2000 or Tcblimit < 4
- 33 Threadlimit exceeds tcblimit or Threadlimit > 2000 or Threadlimit < 3
- 34 Already connected
- 39 DB2 not active
- 40 Insufficient authorization

- 41 Connection error
- 42 Invalid init parms
- 43 DB2ID cannot be set, connection active
- 44 DB2CONN partially discarded
- 46 SET NOTCONNECTED when the FORCE or WAIT option has been specified, but this transaction is itself using the CICS DB2 interface.
- 47 Bad characters in DB2GROUPID
- 48 Both DB2ID and DB2GROUPID specified
- 49 DB2GROUPID cannot be set, connection active
- 50 DB2 module DSNAPRH cannot be found
- 51 TCBLIMIT > MAXOPENTCBS (when connected to DB2 Version 6 or later)
- 52 DB2GROUPID not found
- 53 DB2ID not found
- 54 Invalid RESYNCMEMBER option
- 57 REUSELIMIT > 10000 or REUSELIMIT < 0

SET DB2ENTRY

Sets the attributes of a particular DB2ENTRY used to define resources to be used by a specific transaction or by a group of transactions when accessing DB2.

SET DB2ENTRY

NONE

No accounting records are required for transactions using pool threads. DB2 produces at least one accounting record for each thread when the thread is terminated. Authorization changes additionally cause accounting records to be produced.

TXID The CICS DB2 attachment facility causes an accounting record to be produced when the transid using the thread changes.

Because pool threads are typically used by a number of different transaction IDs, there is an increased chance that a transaction containing multiple units of work (UOWs) will use a different thread for each UOW (assuming the thread is released at syncpoint). In this case an accounting record may be produced per UOW.

TASK The CICS DB2 attachment facility causes a minimum of one accounting record for each CICS task to be produced.

A transaction containing multiple UOWs (assuming the thread is released at syncpoint) may use a different thread for each of its UOWs. The result may be an accounting record produced for each UOW.

UOW The CICS DB2 attachment facility causes an accounting record to be produced for each UOW, assuming that the thread is released at the end of the UOW.

AUTHID

specifies the id to be used for security checking when using this DB2ENTRY. If AUTHID is specified, AUTHType may not be specified.

AUTHTYPE

returns the type of id that can be used for security checking when using this DB2ENTRY. If AUTHType is specified, AUTHID may not be specified. CVDA values are:

GROUP

Specifies the 8-character USERID and the connected group name as the authorization ID. The following table shows how these two values are interpreted by DB2.

IDs passed to DB2	How DB2 interprets values
CICS sign-on user ID (USERID)	Represents the primary DB2 authorization ID.
RACF-connected group name	If the RACF list of group options is not active, DB2 uses the connected group name supplied by the CICS attachment facility as the secondary DB2 authorization ID. If the RACF list of group options is active, DB2 ignores the connected group name supplied by the CICS attachment facility, but the value appears in the DB2 list of secondary DB2 authorization IDs.

To use the GROUP option the CICS system must have RACF external security SEC=YES specified in the CICS system initialization table (SIT).

If no RACF group ID is available for this USERID, an 8-character field of blanks is passed to DB2 as the group ID.

SIGN Specifies the SIGNID parameter of the DB2CONN should be used as the resource authorization ID.

TERM Specifies the terminal identification (four characters padded to eight) as

an authorization ID. An authorization ID cannot be obtained in this manner if a terminal is not connected with the transaction.

If a transaction is started (using a CICS command) and has no terminal associated with it, AUTHTYPE(TERM) should not be used.

TX Specifies the transaction identification (four characters padded to eight) as the authorization ID.

OPID The operator identification associated with the CICS transaction is used as the authorization ID (three characters padded to eight).

USERID

The 8-character USERID associated with the CICS transaction is used as the authorization ID.

When the DB2 sample sign-on exit DSN3@.SGN is used with AUTHTYPE(USERID), the exit sends the user ID to DB2 as the primary authorization ID and the RACF group ID to DB2 as the secondary ID. When the sample sign-on exit is used, there is no difference between AUTHTYPE(USERID) and AUTHTYPE(GROUP).

BUSY(Cvda)

specifies what CICS is to do if a SET DB2ENTRY DISABLED is issued and the entry is busy when the set command is issued. CVDA values are:

WAIT CICS must wait for all activity on the DB2ENTRY to be quiesced before setting the DB2ENTRY disabled. CICS then returns control to the application.

Note that when a DB2ENTRY is quiescing, all existing transactions are allowed to complete. Transactions already queued against the entry are also allowed to complete. New transactions that try to access the DB2ENTRY are routed to the POOL, or abended, or sent a SQLCODE depending on the setting of DISABLEDACT.

NOWAIT

is the same as WAIT except that control returns to the application as soon as the SET DISABLED request is queued.

FORCE

all tasks using the DB2ENTRY, and those queued against the DB2ENTRY are forcepurged. The DB2ENTRY is then DISABLED and control returns to the application.

DISABLEDACT

specifies what CICS is to do with new transactions that access a DB2ENTRY when it has been disabled or disabling. CVDA values are:

POOL The CICS DB2 attachment facility routes the request to the pool. Message DFHDB2072 is sent to the transient data destination specified by MSGQUEUE on the DB2CONN for each transaction routed to the pool.

ABEND

The CICS DB2 attachment facility abends the transaction. The abend code is AD26.

SQLCODE

An SQLCODE is returned to the application indicating that the DB2ENTRY is disabled.

ENABLESTATUS(cvda)

specifies whether the DB2ENTRY can be accessed by applications. CVDA values are:

ENABLED

The DB2ENTRY can be accessed by applications.

DISABLED

The DB2ENTRY cannot be accessed by applications. A DB2ENTRY has to be disabled before it can be reinstalled or discarded.

PLAN

specifies the name of the plan to be used for this DB2ENTRY.

If PLAN is specified, PLANEXITNAME cannot be specified.

PLANEXITNAME

specifies the name of the dynamic plan exit to be used for this DB2ENTRY. If you change the PLAN and PLANExitname while there are active transactions for the DB2ENTRY the next time the transaction releases the thread, the plan/exit is determined using the new rules. If PLANExitname is specified, PLAN cannot be specified.

PRIORITY

specifies the priority of the thread TCBs for this DB2ENTRY relative to the CICS main TCB (QR TCB). The thread TCBs are CICS open L8 TCBs. CVDA values are:

HIGH Thread TCBs have a higher priority than the CICS QR TCB.

EQUAL

Thread TCBs have equal priority with the CICS QR TCB.

LOW Thread TCBs have a lower priority than the CICS QR TCB.

PROTECTNUM

specifies the maximum number of protected threads for this DB2ENTRY.

THREADLIMIT

specifies the maximum number of threads for this DB2ENTRY that the CICS DB2 attachment allows active before requests are made to wait or are rejected.

THREADWAIT

specifies whether or not transactions should wait for a DB2ENTRY thread, be abended, or overflow to the pool should the number of active DB2ENTRY threads reach the THREADLimit number. CVDA values are:

TWAIT

If all threads are busy, a transaction waits until one becomes available.

NOTWAIT

If any threads are busy, a transaction is terminated with an abend code AD2P.

TPOOL

If all threads are busy, the transaction is diverted to use the pool of threads. If the pool is also busy, and NOTWAIT has been specified for the THREADWAIT parameter on the DB2CONN. The transaction is terminated with abend code AD3T.

Notes:

1. When you change the value of AUTHID or AUTHType, a surrogate user security check is invoked if security is active. This ensures that the userid under which SET is being executed is authorized to act on behalf of the userid being set.
2. All parameters on SET DB2ENTRY can be set while the CICS DB2 attachment is active and the transactions are active.

Conditions

NOTAUTH

The user associated with the issuing task is not authorized to use this command, or is not authorized to access this resource.

RESP2 values:

- | | |
|-----|---------------------------------|
| 100 | Command authorization failure |
| 101 | Resource authorization failure |
| 102 | Surrogate authorization failure |
| 103 | Authtype authorization failure |

NOTFND

RESP2 values:

- | | |
|---|---|
| 1 | There is no DB2ENTRY currently installed with the specified name. |
|---|---|

INVREQ

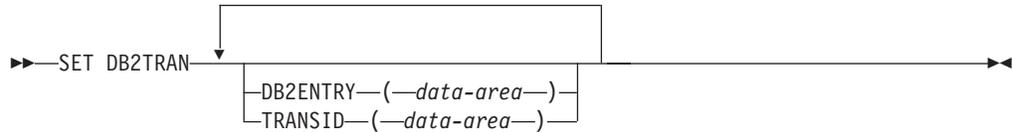
RESP2 values:

- | | |
|----|---|
| 2 | Invalid action value |
| 3 | Invalid Authtype value |
| 4 | Invalid busy value |
| 5 | Invalid enablestatus value |
| 7 | Invalid priority value |
| 8 | Invalid Threadwait value |
| 9 | Bad characters in Authid |
| 10 | Bad characters in Plan |
| 11 | Bad characters in Planexitname |
| 12 | Both Authid and Authtype specified |
| 13 | Both Plan and Planexitname specified |
| 14 | Entry is disabling |
| 15 | Protectnum greater than Threadlimit or protectnum < 0 or protectnum > 2000 |
| 16 | Threadwait must be tpool with Threadlimit=0 |
| 17 | Threadlimit > 2000 or Threadlimit < 0 or Threadlimit > TCBLIMIT |
| 18 | Invalid Accountrec value |
| 19 | SET DISABLED when the FORCE or WAIT option has been specified, but this transaction is itself using the DB2ENTRY. |

SET DB2TRAN

Sets the attributes of a particular DB2TRAN associated with a DB2ENTRY.

SET DB2TRAN



Conditions: NOTAUTH, NOTFND, INVREQ

This command is threadsafe.

Description

A DB2TRAN is identified by the name it was defined with in CEDA. Alternatively, if a transid is specified on a DB2ENTRY when the DB2ENTRY is installed, CICS installs a DB2TRAN named DFHxxxx, where xxxx is the transid.

Options

DB2ENTRY

specifies the name of the DB2ENTRY to which this DB2TRAN refers; that is, the DB2ENTRY with which this additional transid should be associated.

TRANSID

specifies the transaction id to be associated with the entry. You cannot have more than one installed DB2TRAN for the same transaction id. If you specify a transaction id that matches a transaction id specified in an existing installed DB2TRAN, the command will fail. The transaction id can include wildcard characters (see the *CICS Resource Definition Guide* for information about use of wildcard characters). If you change TRANSID for a DB2TRAN while the attachment is active, all transactions with a thread continue to use the thread from that entry until it is released for reuse. When that transaction issues the next SQL statement, the thread is acquired from the entry or pool based upon the new definition.

DB2TRAN parameters may be set at any time.

Conditions

NOTAUTH

The user associated with the issuing task is not authorized to use this command, or is not authorized to access this resource.

RESP2 values:

- 100 Command authorization failure
- 101 Resource authorization failure
- 102 Surrogate authorization failure
- 103 Authtype authorization failure

NOTFND

RESP2 values:

- 1 There is no DB2TRAN currently installed with the specified name.

INVREQ

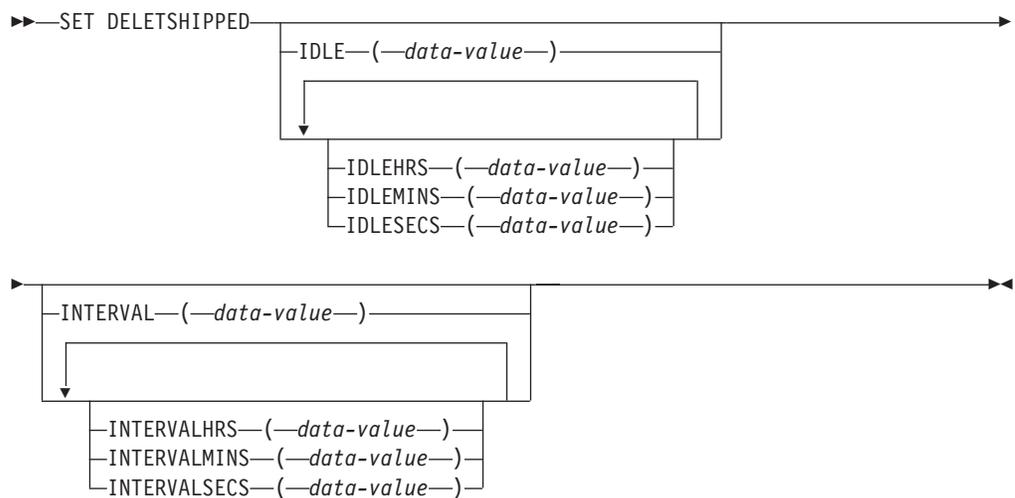
RESP2 values:

- 2 Bad characters in TRANSID name
- 3 Transid already exists in another installed DB2TRAN
- 4 Bad characters in DB2ENTRY name

SET DELETSHIPED

Change the system settings that control automatic deletion of shipped terminal definitions.

SET DELETSHIPED



Conditions: INVREQ, NOTAUTH

Description

The **SET DELETSHIPED** command allows you to change values that control the timeout mechanism that CICS provides for deleting definitions of shipped terminals that are inactive. A shipped definition is inactive if the terminal has not been used locally for a specified period of time and no task that requires the terminal is waiting to be attached. For more information about shipped definitions, see Intercommunication overview in the Intercommunication Guide and **TERMINAL** resources in the Resource Definition Guide.

You can change both the length of time a shipped terminal must remain inactive before being eligible for deletion (**IDLE** time), and the interval at which CICS checks for such terminals (the **INTERVAL**). Time values can be expressed in several different ways:

- A 4-byte packed decimal composite, in the format *Ohhmmss+*, where the hours (*hh*) are in the range 0–99, and minutes (*mm*) and seconds (*ss*) are both from 0–59. Use the **IDLE** and **INTERVAL** options for this format.
- With separate values for hours, minutes, and seconds. Use **IDLEHRS**, **IDLEMINS**, and **IDLESECS** instead of **IDLE** for this format, and **INTERVALHRS**, **INTERVALMINS**, and **INTERVALSECS** instead of **INTERVAL**. You can use any

combination of hours, minutes, and seconds. If you use only one, the time value must be *less* than 100 hours, so that the range for hours is 0-99, the range for minutes is 0-5999, and the range for seconds is 0-359999. If you use two or three, the range is the same for hours, but minutes and seconds must both be in the range 0-59.

For example, to specify an IDLE time of 1 hour and 15 minutes, you could use any of the following:

- IDLE(011500)
- IDLEHRS(1) IDLEMINS(15)
- IDLEMINS(75)
- IDLESECS(4500).

Options

IDLE(*data-value*)

specifies the idle time, as a 4-byte packed decimal value in the form "0hhmmss+". Idle time is the minimum time that a terminal must be inactive to be eligible for deletion.

See the notes at the beginning of this command description for the range of values allowed.

IDLEHRS(*data-value*)

Specifies, as a fullword binary value, the idle time in hours (when used alone) or the hours component of the idle time (when used with IDLEMINS or IDLESECS). See the IDLE option.

IDLEMINS(*data-value*)

Specifies, as a fullword binary value, the idle time in minutes (when used alone) or the minutes component of the idle time (when used with IDLEHRS or IDLESECS). See the IDLE option.

IDLESECS(*data-value*)

Specifies, as a fullword binary value, the idle time in seconds (when used alone) or the seconds component of the idle time (when used with IDLEHRS or IDLEMINS). See the IDLE option.

INTERVAL(*data-value*)

Specifies, as a 4-byte packed decimal value in the form "0hhmmss+", the interval between invocations of the timeout delete mechanism.

When you change the checking interval, the next interval is measured from *the time the command is issued*, **not** from the previous invocation or CICS startup. If you want immediate deletion, use the "PERFORM DELETSHIPPED" on page 585 command.

See the notes at the beginning of this command description for the range of values allowed.

INTERVALHRS(*data-value*)

Specifies, as a fullword binary value, the invocation interval in hours (when used alone) or the hours component of the interval (when used with IDLEMINS or IDLESECS). See the INTERVAL option.

INTERVALMINS(*data-value*)

Specifies, as a fullword binary value, the invocation interval in minutes (when used alone) or the minutes component of the interval (when used with INTERVALHRS or INTERVALSECS). See the INTERVAL option.

INTERVALSECS(*data-value*)

Specifies, as a fullword binary value, the invocation interval in seconds (when used alone) or the seconds component of the interval (when used with INTERVALHRS or INTERVLMINS). See the INTERVAL option.

Conditions

INVREQ

RESP2 values:

- 1 The INTERVAL value is invalid.
- 2 The INTERVALHRS value is not in the range 0-99.
- 3 The INTERVLMINS value is invalid.
- 4 The INTERVALSECS value is invalid.
- 5 The IDLE value is invalid.
- 6 The IDLEHRS value is not in the range 0-99.
- 7 The IDLEMINS value is invalid.
- 8 The IDLESECS value is invalid.

NOTAUTH

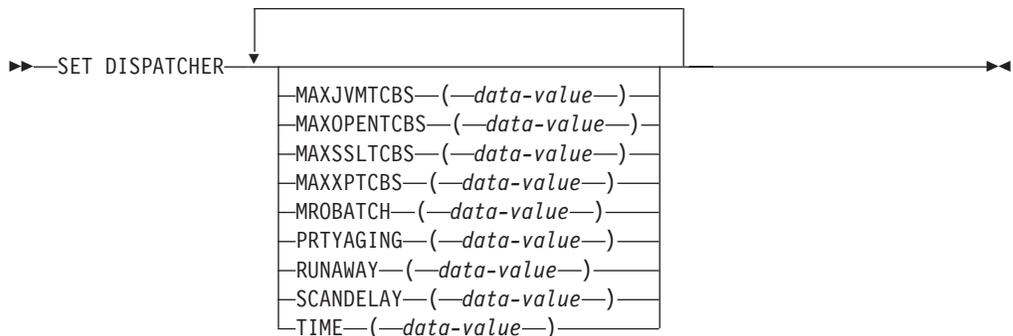
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SET DISPATCHER

Change CICS dispatcher system information.

SET DISPATCHER



Conditions: INVREQ, NOTAUTH,

Description

Use the **SET DISPATCHER** command to change the values of some of the options that CICS dispatcher domain uses for task and TCB management.

These values are set initially by system initialization parameters, described in the *CICS System Definition Guide*. System initialization parameters that correspond to

those in this command have the same or similar names. "INQUIRE SYSTEM" on page 440 lists the exact correspondence.

Options

MAXJVMTCBS (*data-value*)

Specifies, as a fullword binary value, the maximum number of J8 and J9 mode open TCBs that can exist concurrently in the CICS region. The value specified can be in the range 1 to 999. Within this limit, there are no constraints on how many of the TCBs in the JVM pool are J8 TCBs and how many are J9 TCBs.

If you reduce MAXJVMTCBS from its previously defined value, and the new value is less than the number of open TCBs currently allocated, CICS detaches TCBs to achieve the new limit only when they are freed by user tasks.

Transactions are not abended to allow TCBs to be detached to achieve the new limit.

If tasks are queued waiting for a J8 or J9 mode TCB and you increase MAXJVMTCBS from its previously defined value, CICS attaches a new TCB to resume each queued task, up to the new limit.

MAXOPENTCBS (*data-value*)

Specifies, as a fullword binary value, the maximum number of L8 and L9 mode open TCBs that can exist concurrently in the CICS region. The value specified can be in the range 1 to 2000.

If you reduce MAXOPENTCBS from its previously defined value, and the new value is less than the number of open TCBs currently allocated, CICS detaches TCBs to achieve the new limit only when they are freed by user tasks.

Transactions are not abended to allow TCBs to be detached to achieve the new limit.

If tasks are queued waiting for an L8 or an L9 mode TCB, and you increase MAXOPENTCBS from its previously defined value, CICS attaches a new TCB to resume each queued task, up to the new limit.

MAXSSLTCBS (*data-value*)

Specifies, as a fullword binary value, the maximum number of S8 mode open TCBs that can exist concurrently in the CICS region. The value specified can be in the range of 1 - 1024.

MAXXPTCBS (*data-value*)

Specifies, as a fullword binary value, the maximum number of X8 and X9 mode open TCBs that can exist concurrently in the CICS region. The value specified can be in the range 1 - 999.

If you reduce MAXXPTCBS from its previously defined value, and the new value is less than the number of open TCBs currently allocated, CICS detaches TCBs to achieve the new limit only when they are freed by user tasks.

Transactions are not abended to allow TCBs to be detached to achieve the new limit.

If there are tasks queued waiting for an X8 or an X9 mode TCB, and you increase MAXXPTCBS from its previously defined value, CICS attaches a new TCB to resume each queued task, up to the new limit.

MROBATCH (*data-value*)

Specifies, as a fullword binary value, the number of events that must occur, from a list of MRO and DASD I/O events on which CICS is waiting, before CICS is posted explicitly to process them. The value must be in the range 1 - 255.

PRTYAGING (*data-value*)

Specifies, as a fullword binary value, the rate at which CICS is to increase the priority of a task waiting for dispatch. CICS increases the task priority by 1 after each PRTYAGING milliseconds of wait time without a dispatch. The value must be in the range 0 - 65535.

RUNAWAY (*data-value*)

Specifies, as a fullword binary value, the default for runaway task time. This global value for the CICS region is used for any task running a transaction that does not specify an explicit runaway task time.

The value must be either zero, which means that runaway task detection is not required for tasks using the default value, or in the range 500 - 2 700 000. The value you supply is rounded down to the nearest multiple of 500.

SCANDELAY (*data-value*)

Specifies, as a fullword binary value, the maximum number of milliseconds between a user task making a terminal I/O request and CICS dispatching the terminal control task to process it. This value is sometimes called the “terminal scan delay”, and is set initially by the **ICVTSD** system initialization parameter. The value must be in the range 0 - 5000.

TIME (*data-value*)

Specifies, as a fullword binary value, the maximum interval in milliseconds for which CICS gives control to the operating system if no tasks are ready for dispatch. This value is set initially by the **ICV** system initialization parameter and is sometimes called the “region exit time interval”. The TIME value must be in the range 100 - 3 600 000 and must not be less than the SCANDELAY value. You can determine the current SCANDELAY value, if you are not setting it at the same time, with the **INQUIRE DISPATCHER SCANDELAY** command.

Conditions**INVREQ**

RESP2 values:

- 5 The TIME value is not in the range 100 - 3 600 000.
- 6 The RUNAWAY value is out of range.
- 7 MROBATCH is not in the range 1 - 255.
- 13 TIME is less than SCANDELAY.
- 14 PRTYAGING is not in the range 0 - 65535.
- 15 SCANDELAY is not in the range 0 - 5000.
- 26 The MAXOPENTCBS value is less than the TCBLIMIT on the DB2CONN resource definition when CICS is connected to DB2.
- 27 The MAXOPENTCBS value is out of range.
- 28 The MAXJVMTCBS value is out of range.
- 30 The MAXSSLTCBS value is out of range.
- 31 The MAXXPTCBS value is out of range.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SET DOCTEMPLATE

Refresh the cached copy of a document template installed in your CICS region, or phase in a new copy of a CICS program or exit program that is defined as a document template.

SET DOCTEMPLATE

►► SET DOCTEMPLATE(*data-value*) COPY—(*cvda*—) NEWCOPY ►►

Conditions: INVREQ, NOTFND, NOTAUTH

This command is threadsafe.

Description

The SET DOCTEMPLATE command operates on the specified CICS document template. The COPY(NEWCOPY) option is the only option available on this command.

For document templates in a partitioned data set, CICS file, z/OSUNIX System Services HFS file, temporary storage queue, or transient data queue, the command deletes the copy of the document template which is currently cached by CICS, and replaces it with a new copy. (For templates in a partitioned data set, CICS first performs a BLDL (build list) to obtain the most current directory information, and then rereads the member.)

For document templates that reside in CICS programs (with PROGRAM specified in the DOCTEMPLATE resource definition), the command refreshes the program. It is equivalent to SET PROGRAM PHASEIN for the specified program. Document templates retrieved from programs are not cached by CICS.

For document templates generated by exit programs (with EXITPGM specified in the DOCTEMPLATE resource definition), the command refreshes the exit program. It is equivalent to SET PROGRAM PHASEIN for the specified exit program. When you issue the command, CICS deletes any cached copy of the document template, phases in the new copy of the program, and creates a new cached copy of the document template if the exit program specifies caching. The refreshed exit program can specify a different setting for whether or not caching should take place, and CICS honors the change.

Options

COPY(*cvda*)

refreshes the document template. The CVDA value is:

NEWCOPY

If a cached copy of the document template exists, it is to be deleted. If the document template resides in a CICS program or exit program, a new copy of the program is to be phased in. If caching is required for the document template, a new copy of the document template is to be loaded into the cache.

DOCTEMPLATE(*data-value*)

specifies the 1 to 8-character name of the DOCTEMPLATE resource definition which defines the document template.

Conditions

INVREQ

RESP2 values:

- 2 COPY is specified with an invalid CVDA value.
- 4 The new copy of the document template could not be loaded into the cache.

NOTFND

RESP2 values:

- 1 The DOCTEMPLATE resource definition was not found.
- 3 The member of the partitioned data set specified by the DOCTEMPLATE resource definition was not found.
- 5 The resource specified by the DOCTEMPLATE resource definition was not found.

NOTAUTH

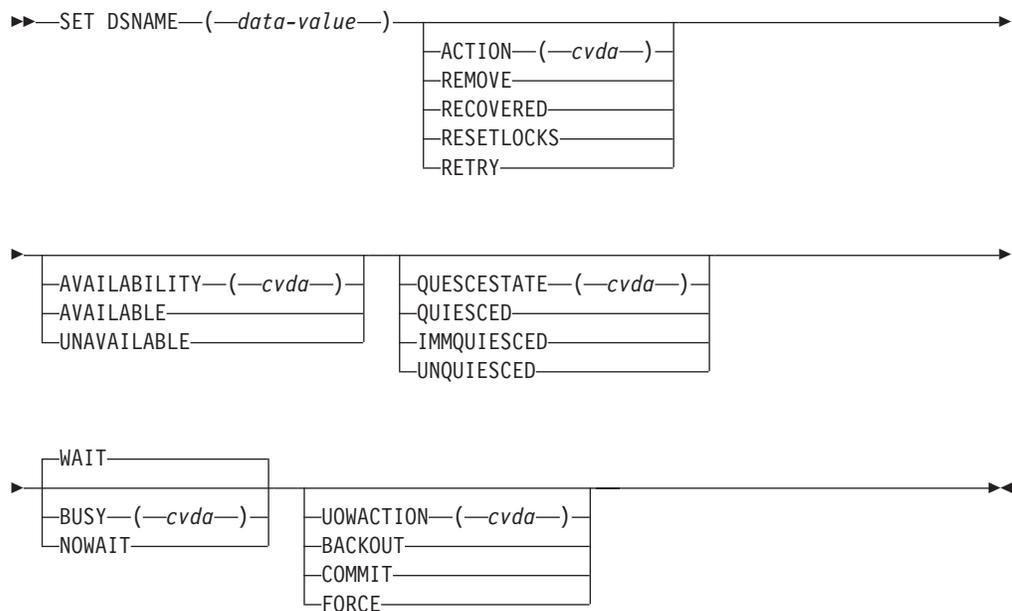
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this DOCTEMPLATE resource definition in the way required by this command.

SET DSNAME

Change information relating to an external data set, including actions that apply to all UOWs that access this data set.

SET DSNAME



Conditions: DSNNOTFOUND, INVREQ, IOERR, NOTAUTH, SUPPRESSED

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

With the SET DSNAME command, you can:

- Tell CICS that a data set is no longer required on the local system.
- Set the “backup while open” (BWO) attributes of the data set to the ‘forward recovered’ state by updating the ICF catalog. This indicates that a forward recovery has taken place.
- Mark a VSAM data set as quiesced, or unquiesced, throughout the sysplex.
- Make a VSAM data set available or unavailable to a CICS region. (The availability function does not operate across the sysplex—a SET DSNAME (...) AVAILABILITY(...) command is effective only within the CICS region in which it is issued.)
- Retry all UOW log records that are shunted due to the failures of this data set (other than indoubt failures).
- Force any UOWs that are shunted due to indoubt failures, and which have updated this data set, to complete.
- Purge shunted UOW log records that hold retained locks (other than those due to indoubt failures) for any records in the data set, and release the retained locks,
- Cancel any attempt to recover lost RLS locks for the data set, using the UOWACTION and ACTION(RESETLOCKS) options.

For information about shunted UOW log records, see the *CICS Recovery and Restart Guide*.

The options and CVDAs for the SET DSNAME command are subject to the following rules relating to the order of processing and the combinations of keywords and multiple keywords on the same command:

- If REMOVE is specified, no other attribute is allowed.
- Options are processed in the following order:
 1. RECOVERED
 2. UNQUIESCED
 3. AVAILABLE
 4. RETRY
 5. UOWACTION
 6. RESETLOCKS
 7. UNAVAILABLE
 8. QUIESCED.

If you specify RETRY, you should not also specify UNAVAILABLE or QUIESCED, because this could cause backout retries to fail.

If you combine UNQUIESCED with any other attributes, also specify BUSY(WAIT), so that later options do not cause the command to fail because the data set is not unquiesced.

Some of the options of a data set cannot be specified until the first file that references the data set has been opened. Where an attribute is not valid until a file

has been opened, the INVREQ condition is returned. QUIESCSTATE is an attribute that can be used before any files have been opened against the specified data set.

Options

ACTION(*cvda*)

specifies the action to be taken on the data set. CVDA values are:

RECOVERED

This data set has been restored from a backup version and forward recovery has been run and completed successfully. CICS attempts to update the BWO attributes of the data set in the ICF catalog using DFSMS™ callable services. The command is used by the database administrator to update the BWO attributes in the ICF catalog if the forward recovery log apply utility does not do so, or if the database administrator finds that there has been no update since the backup copy was made. This would mean that no forward recovery is needed. If the BWO attributes of the data set are not updated after restoring a backup copy, a subsequent file open fails because the data set is still marked as down-level in the ICF catalog.

For information about DFSMS callable services see the *RACF Security Administrator's Guide*.

REMOVE

A data set is no longer required on the local system. Before you can issue a SET DSNAME REMOVE command, the data set must have a FILECOUNT of zero. If you specify REMOVE, you must not specify any other option.

Removing temporary data sets: If you have an application that creates temporary data sets, it is most important that you remove the associated data set name blocks when the data sets are no longer needed. Data set name blocks are not removed when a data set is closed, or when CICS is shut down (they are removed automatically only during a cold or initial start). If not removed, unwanted data set name blocks can use up excessive amounts of dynamic storage, leading to a short-on-storage condition. See "Examples" on page 655 for an illustration of how you can identify and remove unwanted data set name blocks.

RESETLOCKS (VSAM only)

Purges shunted UOW log records for backout-failed and commit-failed UOWs that hold locks on this data set, and releases the retained locks.

- Backout-failed UOWs are those that failed during backout processing.
- Commit-failed UOWs are those that have updated RLS data sets, and have failed to release locks during the second phase of 2-phase commit syncpoint processing.

If you specify this option, you are accepting backout failure and some loss of data integrity rather than retaining locks and delaying transactions, and therefore it should be used only as a last resort.

For backout-failed and commit-failed UOWs that hold locks on the data set, all records relating to this data set are removed from the system log and all retained record locks held by this CICS for the data set are released. Diagnostic messages are written to the CSFL transient

data queue for each backout-failed log record that is removed as a result of the RESETLOCKS operation.

You might choose to use RESETLOCKS if backout-failed or commit-failed log records are holding up lost locks recovery for the data set, and there is no other way of resolving them.

Notes:

- This option does not apply to shunted *indoubt* UOWs. You should try to resolve the shunted *indoubt* UOWs that hold locks on the data set in other ways before issuing RESETLOCKS; for example, by using COMMIT, BACKOUT, or FORCE (see the UOWACTION option).
- RESETLOCKS can fail during the commit phase (for example, if an error occurs while CICS is trying to release the RLS locks), in which case the UOWs revert to being shunted as commit-failed UOWs.

RETRY

Shunted UOW log records, caused by failed backout and commit processing for this data set, should be retried. This is similar in concept to the SET CONNECTION RESYNC command, but applies to backout-failed and commit-failed UOWs only, and not to *indoubt* UOWs.

You should use RETRY when the data set has shunted backout-failed or commit-failed UOWs associated with it, and you believe that some or all of the data set problems are either transient or have been resolved. If the data set was damaged in some way, it must have been repaired (recreated) and made available for RETRY to work successfully.

Messages issued at the time of a data set failure, and which cause UOWs to be shunted, recommend the actions required to recover from the failure.

RETRY does not harm data integrity, and can be used safely at any time to enable some failed recovery work to complete.

AVAILABILITY(*cvda*) (VSAM only)

specifies whether the data set is to be marked, in this CICS region, as available or unavailable for use. This command sets or unsets the availability indicator, which is a local flag that a CICS region maintains in a data set name block (DSNB) for each data set. CVDA values are:

AVAILABLE

The data set is available. CICS can issue both RLS and non-RLS open requests for this data set.

UNAVAILABLE

The data set is unavailable. The data set cannot be opened in either RLS or non-RLS modes.

BUSY(*cvda*) (RLS only)

specifies whether CICS should wait when requested to quiesce or unquiesce the data set, provided QUIESCESTATE has also been specified. It is ignored if QUIESCESTATE is not specified. CVDA values are:

NOWAIT

CICS returns control to the application immediately, having started the

quiesce or unquiesce operation asynchronously. You can use INQUIRE DSNNAME QUIESCESTATE to check whether the quiesce or unquiesce has completed.

WAIT CICS returns control to the application only when the data set has been quiesced or unquiesced throughout the sysplex, or when it has failed to do so. If a quiesce is not completed within the time specified in the QUIESTIM system initialization parameter, the quiesce times out. See QUIESTIM system initialization parameter in the System Definition Guide. If you specify WAIT, or allow it to default, you should ensure that your program handles an AEXY abend in case the DTIMOUT value is not high enough to allow your task to wait for completion.

DSNAME(*data-value*)

specifies the name of the data set. It can be up to 44 characters long, and is defined to CICS in the DSNNAME operand of the CEDA DEFINE FILE command.

QUIESCESTATE(*cvda*) (**RLS only**)

specifies the RLS quiesce state of the data set. The state is set in the ICF catalog entry for the data set when the operation has completed. CVDA values are:

IMMQUIESCED

All existing CICS files open in RLS mode throughout the sysplex are closed and the data set is marked as quiesced in the ICF catalog. Each CICS in the sysplex abends all in-flight UOWs that are accessing the data set before closing files, causing in-flight UOWs to back out. Any UOWs that fail backout are shunted. No files can open in RLS mode against this data set, but non-RLS open requests are permitted (although opens for update are not possible in non-RLS mode if the data set has retained RLS locks).

In addition to closing open files, IMMQUIESCED sets the file state to UNENABLED if it was ENABLED. A subsequent **SET DSNNAME UNQUIESCED** command restores the file state to ENABLED, provided it was set UNENABLED by a QUIESCED or IMMQUIESCED action, but *not* if the UNENABLE state is because of some other event. This state change is recorded in the CICS global catalog.

Note: Using the IMMQUIESCED option causes any tasks currently using the data set to be terminated immediately, using the CICS task FORCEPURGE mechanism. In some extreme cases, CICS may terminate abnormally. For this reason, setting a data set as quiesced using the IMMQUIESCED option should be restricted to exceptional circumstances.

QUIESCED

All existing CICS files open in RLS mode throughout the sysplex are closed and the data set is marked as quiesced in the ICF catalog. Each CICS in the sysplex waits until all in-flight UOWs that are accessing the data set have reached syncpoint before closing the files; that is, the UOWs are:

- successfully committed
- *or* successfully backed out
- *or* shunted because of an indoubt failure
- *or* shunted because of a failed backout
- *or* shunted because of a failed commit.

Note: If you specify QUIESCED with WAIT (the default), all tasks in all CICS regions in the sysplex must have reached syncpoint before the files are closed, allowing your command to complete. You must ensure that the DTIMOUT value for the transaction issuing the QUIESCED command is sufficient to allow for this, otherwise the transaction abends with an AEXY abend. The QUIESCE operation is allowed to run until completed or until the timeout value set by the QUIESTIM system initialization parameter, (for which the default is 4 minutes), is reached.

No files can open in RLS mode against this data set, but non-RLS open requests are permitted (although opens for update are not possible in non-RLS mode if the data set has retained RLS locks).

In addition to closing open files, QUIESCED sets the file state to UNENABLED if it was ENABLED. A subsequent SET DSNAME UNQUIESCED command restores the file state to ENABLED, provided it was set UNENABLED by a QUIESCED or IMMQUIESCED action, but *not* if the UNENABLE state is because of some other event. This state change is recorded in the CICS global catalog.

UNQUIESCED

The data set is marked as unquiesced in the ICF catalog. RLS or non-RLS opens can be issued against this data set, the access mode (RLS or non-RLS) being established by the first open. After the first successful open request, subsequent open requests in the same mode as the first open only are permitted.

If a file has been set UNENABLED by an earlier SET DSNAME IMMQUIESCED or QUIESCED command, UNQUIESCED restores the file state to ENABLED. This state change is recorded in the CICS global catalog.

UOWACTION(*cvda*)

specifies the action to be taken for shunted indoubt UOWs. CVDA values are:

BACKOUT

All shunted indoubt UOWs that hold locks on this data set should be backed out.

COMMIT

All shunted indoubt UOWs that hold locks on this data set should be committed.

FORCE

All shunted indoubt UOWs that hold locks on this data set should be FORCED to back out or commit, as specified by the ACTION attribute defined on the transaction resource definition.

Conditions

DSNNOTFOUND

RESP2 values:

- 1 The named data set cannot be found.
- 15 RECOVERED was specified, but the data set was not found.

INVREQ

RESP2 values:

- 3 ACTION has an invalid CVDA value.

- 10 REMOVE was specified, but the data set is associated with a file definition.
- 12 REMOVE was specified with another option. If you specify REMOVE, it must be the only option present on the command.
- 13 REMOVE was specified but a lock was held on the data set by another INQUIRE or SET DSNNAME command, or by CICS file control processing.
- 14 RECOVERED was specified but CICS is not configured to support "backup while open" (BWO). Check that you have a version of MVS/DFP™, DFHSM, and DFDSS that supports BWO.
- 16 RECOVERED was specified but the data set has not been opened during this CICS session, so the BWO attributes in the ICF catalog cannot be set.
- 17 RECOVERED was specified for a BDAM data set, or a VSAM path. This is not supported.
- 18 RECOVERED was specified for a VSAM base data set that has FCTs open. This is not allowed.
- 19 RECOVERED was specified for an unknown data set, or the data set was not in the 'forward recovered' state.
- 29 QUIESCESTATE is specified, but the operation is not supported because RLS=NO is specified as a system initialization parameter, or because DFSMS 1.3 or later is not installed.
- 30 QUIESCESTATE has an invalid CVDA value.
- 31 BUSY has an invalid CVDA value.
- 33 AVAILABILITY has an invalid CVDA value.
- 34 A QUIESCESTATE value of QUIESCED or IMMQUIESCED is specified, but is rejected by SMSVSAM either because a quiesce or unquiesce is already taking place, or because DFSMSdss is currently taking a backup copy of the data set.
- 36 A QUIESCESTATE value of UNQUIESCED is specified, but is rejected by SMSVSAM either because an unquiesce is already taking place, or because DFSMSdss™ is currently taking a backup copy of the data set.
- 39 AVAILABILITY, QUIESCESTATE, RESETLOCKS, or RETRY is specified for a data set that is a BDAM data set.
- 40 The CICS control block (DSNB) describing the data set has been deleted (by the REMOVE option) by another task before CICS could process this SET command.
- 41 QUIESCESTATE is specified for a data set that is not known to DFSMS as a VSAM data set.
- 42 An invalid CVDA is specified for UOWACTION.
- 43 A QUIESCESTATE value of QUIESCED or IMMQUIESCED is specified without NOWAIT, and the issuing task has updated the data set, or is browsing the data set, in the same unit of work. This is not allowed because:
 - For QUIESCED, this would result in a deadlock.

- For IMMQUIESCED, this would result in the issuing task being purged.
- 44 A SET DSNAME REMOVE command has been issued by another task. This has been detected after this SET DSNAME command was issued, but before the AVAILABILITY option is processed.
- 46 BKOUTSTATUS is specified with a value other than NORMALBKOUT (BKOUTSTATUS is obsolete).
- 47 No file has been opened against the data set since the last cold start of this CICS region, or since the first file definition was installed for the data set.

IOERR

RESP2 values:

- 20 RECOVERED was specified but an error was raised on accessing the ICF catalog. Ensure that the specified data set is on an SMS-managed DASD and is known to the SMS subsystem.
- 21 RECOVERED was specified but an error was raised by the CICS table manager program.
- 35 QUIESCESTATE is specified but the SMSVSAM server is not available.
- 40 QUIESCESTATE is specified, and an unexpected error occurred in DFSMS.
- 48 The specified operation cannot be completed because the data set is migrated. Recall the data set and reissue the command.
- 49 An error was raised by DFSMS when reading the ICF Catalog to establish the base data set name.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SUPPRESSED

RESP2 values:

- 37 A QUIESCESTATE value of QUIESCED or IMMQUIESCED is specified, but the quiesce of the data set is canceled by another participating CICS region. This could be for one of the following reasons:
- A user issued a SET DSNAME UNQUIESCED command.
 - An XFCVSDS global user exit program suppressed the quiesce.
 - An XFCSREQ global user exit program suppressed the close of a file that is open against the data set.
- 38 A QUIESCESTATE value of QUIESCED or IMMQUIESCED is specified, but the quiesce of the data set is canceled by this CICS region because the quiesce operation timed out. This is probably because of a long-running transaction on another participating CICS region preventing the close of a file that is open against the data set.

Another reason for the timeout could be that one or more regions are very busy. If this occurs too frequently, you can modify the timeout period (from the default of 240 seconds) by specifying a longer period using the QUIESTIM system initialization parameter.

Examples

It is possible in CICS to create VSAM data sets online for temporary use, and which are dynamically allocated by CICS file control. Typically, this involves reusing the same file control entry and setting the new temporary data set name each time you need to use a new data set. This practice can lead to a large number of data set name blocks occupying CICS dynamic storage. These can only be removed by a SET DSNAME(...) REMOVE command, or by a cold or initial start of CICS.

Ideally, an application that creates and uses a temporary data set should explicitly delete the DSN block when it no longer needs the data set. This involves two actions:

1. Breaking the association between the CICS file and the data set by issuing an EXEC CICS SET FILE(...) CLOSED DISABLED command, followed by an EXEC CICS SET FILE command to set the DSNAME operand to a null value.
2. Removing the data set name block by issuing an EXEC CICS SET DSNAME(...) REMOVE command.

To set the DSNAME to null you must code the CICS commands as shown in the following examples to ensure they translate and compile correctly.

Assembler example

```
*          Remove DSN block from CICS storage
*
*ASM XOPTS(SP)
DFHEISTG DSECT
TEMPDSN DS    CL44
REMOVE    CSECT
          PRINT GEN
*          Find name of temporary data set if not known
EXEC CICS INQUIRE FILE('TEMPFILE') DSNAME(TEMPDSN)
*          Close file temporary file and set DSN to null
EXEC CICS SET FILE('TEMPFILE') CLOSED DISABLED
EXEC CICS SET FILE('TEMPFILE') DSNAME(=X'00')
*          Remove DSN block from storage
EXEC CICS SET DSNAME(TEMPDSN) REMOVE
*
*          Return and end
*
RETURN    DS    0H
          EXEC CICS RETURN
          END
```

PL/I example

```
*PROCESS XOPTS(SP);
  REMOVE:PROC OPTIONS(MAIN);
  DCL PLIXOPT STATIC EXTERNAL CHAR(10) VAR INIT('NOSTAE');
  DCL TEMPDSN CHAR(44);
/*                                     */
/*          Find name of temporary data set if not known */
EXEC CICS INQUIRE FILE('TEMPFILE') DSNAME(TEMPDSN);
*/
/*          Close file temporary file and set DSN to null */
EXEC CICS SET FILE('TEMPFILE') CLOSED DISABLED;
EXEC CICS SET FILE('TEMPFILE') DSNAME('00'X);
/*          Remove DSN block from storage */
EXEC CICS SET DSNAME(TEMPDSN) REMOVE
/*
```

```

/*          Return and end          */
/*
EXEC CICS RETURN;
END;

```

COBOL example

```

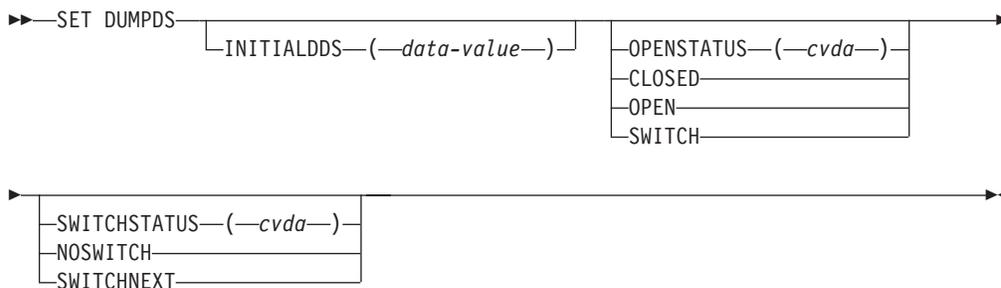
CBL XOPTS(SP)
  IDENTIFICATION DIVISION.
  PROGRAM-ID. REMOVE.
  ENVIRONMENT DIVISION.
  DATA DIVISION.
  WORKING-STORAGE SECTION.
  77 TEMPDSN          PIC X(44).
  PROCEDURE DIVISION.
  * Find name of temporary data set if not known
  EXEC CICS INQUIRE FILE('TEMPFILE') DSNAME(TEMPDSN)
  END-EXEC.
  * Close file temporary file and set DSN to null
  EXEC CICS SET FILE('TEMPFILE') CLOSED DISABLED
  END-EXEC.
  EXEC CICS SET FILE('TEMPFILE') DSNAME(LOW-VALUES)
  END-EXEC.
  * Remove DSN block from storage
  EXEC CICS SET DSNAME(TEMPDSN) REMOVE END-EXEC.
  *
  * Return and end
  EXEC CICS RETURN END-EXEC.
  GOBACK.

```

SET DUMPDS

Change the status of the transaction dump data sets.

SET DUMPDS



Conditions: INVREQ, IOERR, NOTAUTH

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The SET DUMPDS command allows you to change the status of CICS transaction dump data sets. Normally, either there is one of these, known as the ‘A’ dump data set, or there are two, ‘A’ and ‘B’. One is “active” (receiving dumps) and the other, if there are two, is “inactive” (standby). Specifically, you can:

- Open or close the active data set.
- Switch the roles of the active and standby data sets.

- Request CICS to switch automatically when the active data set is full.
- Specify which data set is active the next time CICS is initialized.

Note: If a CICS system is initialized without any transaction dump data sets, only the last two functions are available.

Control does not return to the task issuing the command until the requested change has been made.

Options

INITIALDDS(*data-value*)

specifies, as a 1-character value, which dump data set is to be active first on subsequent warm or emergency restarts. This value is recorded in the CICS global catalog and overrides the previous value, which is set initially by the DUMPDS system initialization option.

The values permitted are A, B, and X. X means that CICS is to use the data set that was not active when CICS last terminated (normally or abnormally); it corresponds to the AUTO setting for the DUMPDS option. (See the *CICS System Definition Guide* for a description of this option.)

OPENSTATUS(*cvda*)

specifies actions to be taken on the transaction dump data sets. CVDA values are:

CLOSED

The active CICS dump data set is to be closed.

OPEN The active CICS dump data set is to be opened.

SWITCH

The roles of the dump data sets are to be switched, if there are two. The data set that is currently active is to become standby, and closed if it is open. The current standby is to become the active data set, and opened if closed.

If you attempt to change the open status of a data set that does not exist, an IOERR exception condition occurs. This can happen if you specify SWITCH when there is only one dump data set, or if you specify any OPENSTATUS value when there are no dump data sets.

SWITCHSTATUS(*cvda*)

specifies whether CICS is to switch active data sets automatically the next time the current dump data set fills. The SWITCHSTATUS value is recorded in the CICS global catalog, and therefore is remembered over warm and emergency restarts. (It is set initially by the DUMPSW system initialization option, described in the *CICS System Definition Guide*.) An automatic switch occurs only once; another SET DUMPDS SWITCHNEXT command is required after each switch to maintain automatic switching. CVDA values are:

NOSWITCH

The data sets are not be switched.

SWITCHNEXT

The data sets are to be switched. (SWITCHNEXT has no effect unless there are two dump data sets at the time the active one fills.)

Conditions

INVREQ

RESP2 values:

- 1 INITIALDDS has an invalid value.
- 2 SWITCHSTATUS has an invalid CVDA value.
- 3 OPENSTATUS has an invalid CVDA value.

IOERR

RESP2 values:

- 4 OPEN or SWITCH caused an error opening a data set.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

Examples

```
EXEC CICS SET DUMPDS  
            INITIALDDS('A')  
            SWITCH  
            NOSWITCH
```

This example tells CICS that the A dump data set is to be active first on subsequent warm and emergency restarts. The OPENSTATUS setting of SWITCH makes the currently active dump data set inactive, and the currently inactive dump data set active. The NOSWITCH option tells CICS that when the (new) active dump data set is full, there is to be no automatic switch to the inactive dump data set.

SET ENQMODEL

Change the status of an ENQMODEL definition.

SET ENQMODEL

```
►► SET ENQMODEL—(—data-value—) STATUS—(—cvda—) ►►  
    DISABLED  
    ENABLED
```

Conditions: INVREQ, NOTAUTH, NOTFND

Description

The SET ENQMODEL command allows you to ENABLE or DISABLE ENQMODEL resources installed on the local system. An ENQMODEL must be enabled to allow matching EXEC ENQ requests to be processed. It must be disabled to allow a more specific ENQMODEL to be enabled.

ENQMODELs forming nested generic enqnames must be enabled in order, from the most to the least specific. For example, enable ABCD* then ABC* then AB*. If you attempt to enable a more specific ENQMODEL when a less specific enqmodel is already enabled, the result is that msg NQ0107 is issued and INVREQ is returned to the caller.

In this case you may need to disable one or more less specific ENQMODELS to allow a more specific ENQMODEL to be enabled. You will then be able to enable the less specific ENQMODELS again.

You cannot enable/disable an ENQMODEL which is in the waiting state. If attempted, INVREQ is returned to the caller.

Options

ENQMODEL(*data-value*)

specifies the 8-character identifier of the resource definition.

STATUS(*cvda*)

specifies the action to be taken on the ENQMODEL. CVDA values are:

ENABLED

If the ENQMODEL is DISABLED, it is ENABLED. Once enabled, matching ENQ requests are processed in the normal way.

DISABLED

The ENQMODEL is put into the WAITING state until there are no enqueues in the local system which match the ENQNAME pattern. It will then be DISABLED. Once disabled, matching ENQ requests will be rejected, and the issuing task is abended.

Conditions

INVREQ

RESP2 values:

- 2 The attempt to enable/disable an ENQMODEL failed, because a more generic ENQMODEL is enabled.
- 3 STATE has an invalid CVDA value.
- 4 The ENQMODEL is in the WAITING state

NOTAUTH

RESP2 values:

- 100 The user of the issuing task is not authorized to use this command.

NOTFND

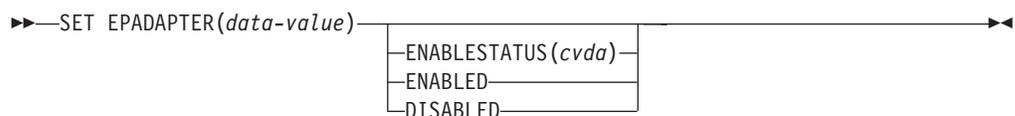
RESP2 values:

- 1 The specified ENQMODEL is not installed on this system.

SET EPADAPTER

Set the status of a specified EP adapter to enabled or disabled.

SET EPADAPTER



Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the SET EPADAPTER command to change the status of a particular EP adapter. Changing the status of an EPADAPTER has no effect on the status of any related EVENTBINDINGS.

Options

EPADAPTER (*data-value*)

Specifies the 32-character name of an EP adapter.

ENABLESTATUS (*cvda*)

Specifies whether events are to be dispatched to this EP adapter. The CVDA values are as follows:

ENABLED

The event processing dispatcher can dispatch events to this EP adapter. The event processing dispatcher starts dispatching events to this EP adapter immediately.

DISABLED

The event processing dispatcher should stop dispatching events to this EP adapter. The event processing dispatcher stops dispatching events to this EP adapter immediately. Any events already dispatched to the EP adapter are emitted.

Conditions

INVREQ

RESP2 values are:

4 ENABLESTATUS has an invalid CVDA value.

NOTAUTH

RESP2 values are:

100 The user associated with the issuing task is not authorized to use this command.

101 The user associated with the issuing task is not authorized to update the EP adapter.

NOTFND

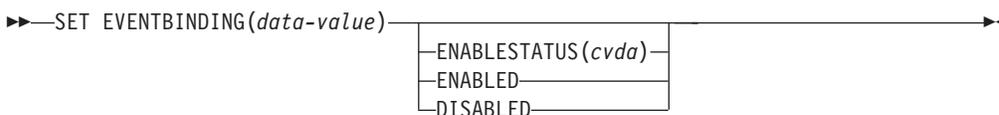
RESP2 values are:

3 The specified EP adapter cannot be found.

SET EVENTBINDING

Set the status of a specified event binding to enabled or disabled.

SET EVENTBINDING



Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

Use the SET EVENTBINDING command to change the status of a particular event binding. If installation of the EVENTBINDING caused an EPADAPTER of the same name to also be installed, the state of the related EPADAPTER, if still available, is unaffected.

Options

EVENTBINDING (*data-value*)

Specifies the 32-character name of an event binding.

ENABLESTATUS (*cvda*)

Specifies whether events matching capture specifications in this event binding are captured and emitted. CVDA values are as follows:

ENABLED

The event binding is enabled. Capture of events matching capture specifications in this event binding starts immediately.

DISABLED

The event binding is disabled. Capture of events matching capture specifications in this event binding is stopped immediately. Any events already captured are emitted.

Conditions

INVREQ

RESP2 values are:

- 4 ENABLESTATUS has an invalid CVDA value.

NOTAUTH

RESP2 values are:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to update the event binding.

NOTFND

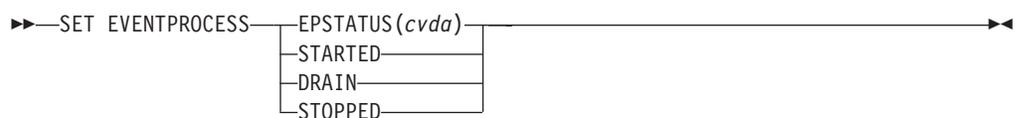
RESP2 values are:

- 3 The specified event binding cannot be found.

SET EVENTPROCESS

Set the status of event processing.

SET EVENTPROCESS



Conditions: INVREQ, NOTAUTH

This command is threadsafe.

Description

Use the SET EVENTPROCESS command to change the status of event processing.

Note: Do not change the status of event processing (that is, set to start, drain, or stop) while a unit of work that captures synchronous transactional events is in progress because you might cause the events to be backed out and the transaction to end abnormally.

Options

EPSTATUS(*cvda*)

a CVDA value changing the current status of event processing.

STARTED

The EVENTPROCESS state is changed to STARTED. For in-flight transactions, the capture of non-transactional events starts immediately and the capture of transactional events starts at the next sync point.

DRAIN

The EVENTPROCESS state is changed to DRAIN, and event capture is stopped immediately.

Any transactional events on the dispatcher queue will be deleted. Transactional events are not considered captured until a syncpoint occurs and a syncpoint event will not now be captured.

Any non-transactional events will be emitted. When the last event on the queue is emitted, the EVENTPROCESS state changes to STOPPED.

STOPPED

The EVENTPROCESS state is changed to STOPPED, and event capture is stopped immediately.

All events on the dispatcher queue are deleted.

Conditions

INVREQ

RESP2 values are:

- 4 Event processing cannot be started while it is draining.
- 5 EPSTATUS has an invalid CVDA value.

NOTAUTH

RESP2 values are:

- 100 The user associated with the issuing task is not authorized to use this command.

SET FILE

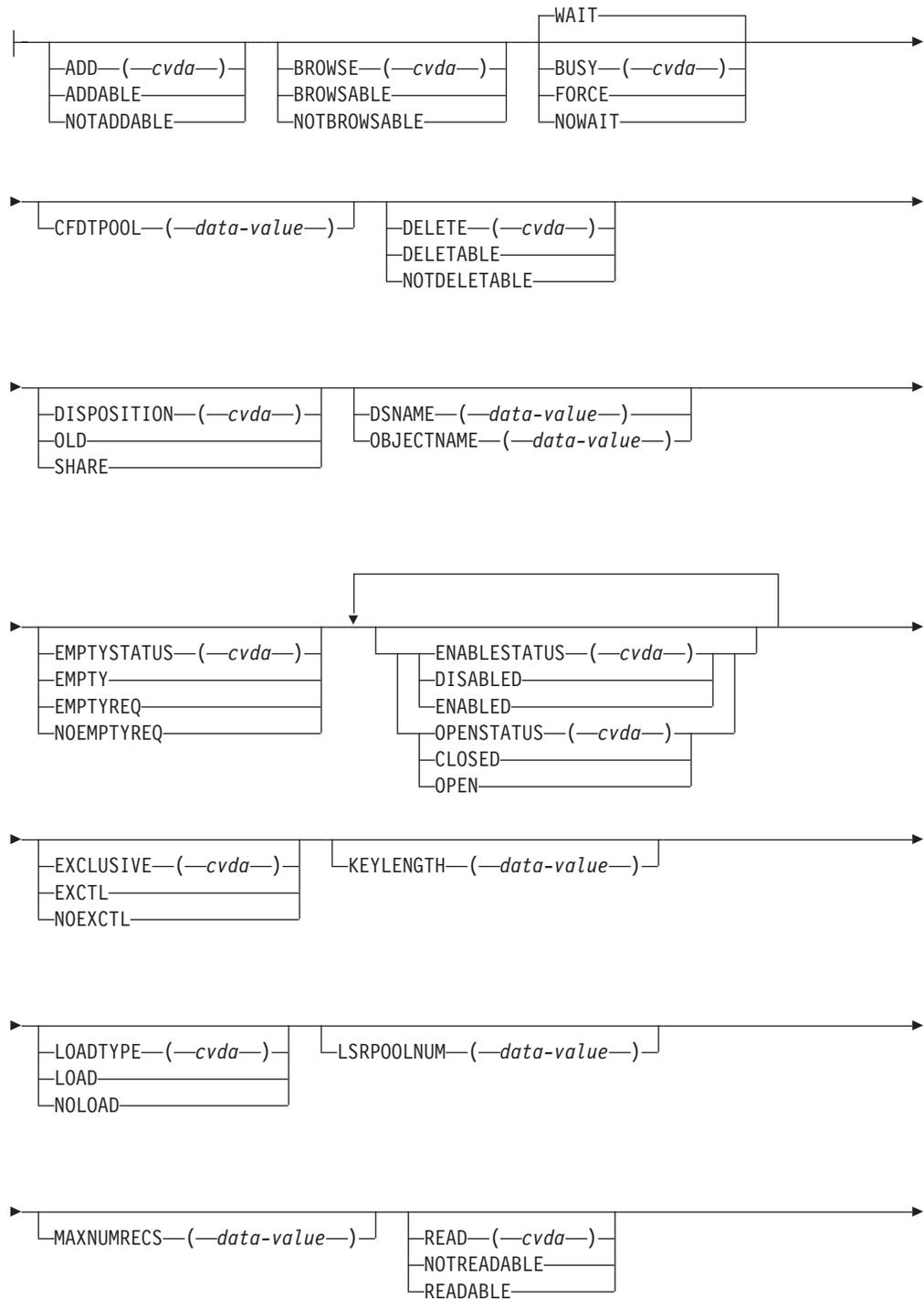
Change attributes of a VSAM or BDAM file, including files that refer to CICS shared data tables and coupling facility data tables.

SET FILE

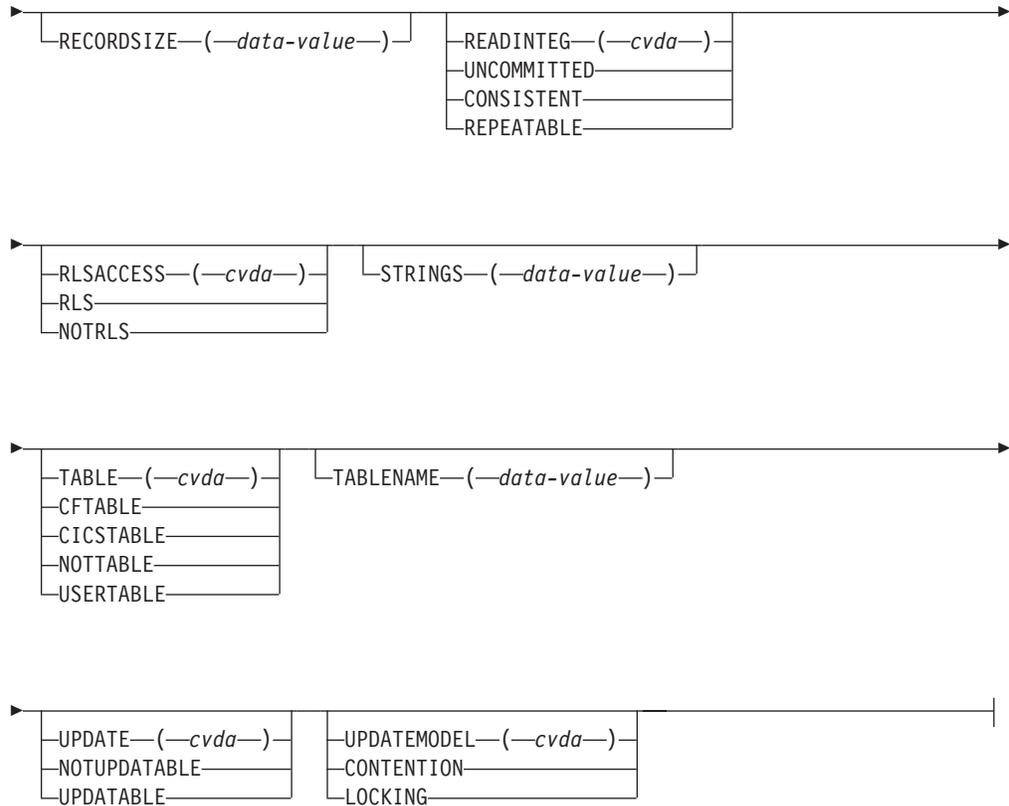
```
▶▶ SET FILE—(—data-value—) | options | ◀◀  
  SET DATASET—(—data-value—)
```

Conditions: FILENOTFOUND, INVREQ, IOERR, NOTAUTH

Options:



1



For more information about the use of CVDAs, see CICS-value data areas (CVDA) in CICS System Programming Reference.

Note: This command replaces the **SET DATASET** command. The **DATASET** keyword is supported by the translator as a synonym for **FILE**, but use **FILE** for all new applications. Similarly, **OBJECTNAME** is supported as a synonym for **DSNAME**.

Any combination of the options can be set on one command. For all changes, other than to close and disable the file, the file must be in a **CLOSED** state, with an **ENABLESTATUS** of either **DISABLED** or **UNENABLED**. Changes do not take effect until the file is next opened.

You can use the **SET FILE** command to set combinations of attributes that are relevant to more than one file type, to simplify switching between different types of file. Attributes that are not relevant to the current type of file are ignored. So you can set up dual-purpose file definitions, for example, by defining both local and remote attributes, or set attributes that make it easy to switch the file from accessing a user-maintained data table in a single MVS image to accessing a coupling facility data table in a parallel sysplex.

If a coupling facility data table exists, and the table attributes specified on the **SET FILE** command do not match those with which it was created, an attempt to open the file fails with an error message.

If you use the **SET FILE** command to switch the file from referencing a coupling facility data table to a different object (for example from CFTABLE to NOTTABLE), the CFDT is not deleted and remains in its pool (the coupling facility list structure).

The requested changes are applied in the following order: NOEMPTYREQ, CLOSED, DISABLED, miscellaneous, OPEN, ENABLED.

SET FILE ENABLED

▶▶ EXEC CICS SET FILE (—data-value—) ENABLED ▶▶

Conditions: FILENOTFOUND, INVREQ, NOTAUTH

SET FILE DISABLED

▶▶ EXEC CICS SET FILE (—data-value—) DISABLED WAIT
NOWAIT
FORCE ▶▶

Conditions: FILENOTFOUND, INVREQ, NOTAUTH

SET FILE OPEN

▶▶ EXEC CICS SET FILE (—data-value—) OPEN EMPTY ▶▶

Conditions: FILENOTFOUND, IOERR, NOTAUTH

SET FILE CLOSED

▶▶ EXEC CICS SET FILE (—data-value—) CLOSED EMPTY WAIT
NOWAIT
FORCE ▶▶

Conditions: FILENOTFOUND, INVREQ, IOERR, NOTAUTH

Description

Use the **SET FILE** command to change some of the attributes of a named VSAM or BDAM file. A security check is made and an unauthorized command attempt is given a NOTAUTH response. If any retained locks are associated with the file, the only attributes that you can change are the ENABLESTATUS and the

OPENSTATUS. If you try to specify any other attribute when there are retained locks, an INVREQ condition is issued.

Options

ADD(*cvda*)

Specifies whether new records can be added to the file. CVDA values are as follows:

ADDABLE

New records can be added to the file.

NOTADDABLE

New records cannot be added to the file.

BROWSE(*cvda*)

Specifies whether the file can be browsed. CVDA values are as follows:

BROWSABLE

The file can be browsed.

NOTBROWSABLE

The file cannot be browsed.

BUSY(*cvda*)

Specifies the CICS action if the file is in use when you issue the SET command. The BUSY option is valid only for requests to set the file to DISABLED or CLOSED, and is ignored for any other request. CVDA values are as follows:

FORCE

All tasks using the fileabend, the file is immediately set to DISABLED or CLOSED, and control returns to the issuing application.

NOWAIT

The same as WAIT, except that CICS returns control to the issuing application as soon as the SET request has been queued.

WAIT CICS waits until all activity on the file has quiesced before setting the file to DISABLED or CLOSED. CICS then returns control to the application that is issuing this command. WAIT is the default.

Closing a file using the FORCE option causes tasks of any current users of the file to be stopped immediately by the CICS task FORCEPURGE mechanism. Data integrity is not guaranteed with this mechanism. In some extreme cases (for example, if an error occurs during backout processing), CICS might end abnormally. For this reason, close files using the FORCE option only in exceptional circumstances.

CFDTPOOL(*data-value*) (**CFDT only**)

Specifies the name of the pool that contains the coupling facility data table. You can specify the CFDT pool name for a file that does not currently refer to a coupling facility data table, but which might be switched to use a coupling facility data table at a later date.

DELETE(*cvda*) (**VSAM only**)

Specifies whether records can be deleted from the file. CVDA values are as follows:

DELETABLE

Records can be deleted from the file.

NOTDELETABLE

Records cannot be deleted from the file.

DISPOSITION(*cvda*)

Specifies the disposition for this file. When you issue a SET FILE DISPOSITION command, you override the current DISPOSITION value, which can have been taken from the installed file definition, or from any JCL statement for this file, if the file has been opened. CVDA values are as follows:

OLD The disposition value is OLD.

SHARE

The disposition value is SHARE.

DSNAME(*data-value*)

Specifies the data set name of the data set associated with this file, as defined to the access method and the operating system. The name can be up to 44 characters long. If you set a value of blanks, CICS does not change the value of this option.

If no JCL statement exists for this file when it is opened, the open is preceded by a dynamic allocation of the file using this data set name. If there is a JCL statement, it takes precedence over the data set name that is specified in this option.

If the file is associated with a coupling facility data table, DSNAME specifies the name of the source data set from which the table is loaded when the file definition specifies LOAD(YES).

When you add a data set name to a file definition for a coupling facility data table, LSR pool size calculations might be involved when the file is opened. This calculation occurs when the file refers to an LSRPOOL that CICS builds using default values, and the first data set that uses the LSR pool is opened to load the table. CICS issues message DFHFC0208, which indicates that a delay might occur while the LSR pool size is being calculated. If you specify a data set name on a file that refers to an LSR pool that is already built using default values, the data set has not been included in the LSR pool calculation, indicating that the existing LSR pool might not be adequate for the new data set. To resolve any problems associated with an LSR pool used in this way, you can close all files that refer to the pool, which causes CICS to discard the pool and rebuild it using new calculations the next time a file is opened that refers to the pool. Alternatively, define the LSR pool explicitly specifying the appropriate values. See LSRPOOL resources the *CICS Resource Definition Guide* for information about defining LSR pools.

With the SET FILE command, you can dissociate the file from any DSNAME by supplying a DSNAME value that begins with a null character (hexadecimal zeros).

EMPTY

Is equivalent to EXEC CICS SET FILE EMPTYSTATUS(EMPTYREQ). It is supported for compatibility reasons only, but not for files operating in RLS mode.

EMPTYSTATUS(*cvda*) (**VSAM only**)

Specifies whether the data set is to be emptied when a file that refers to it is next opened. This option is valid only for data sets that are defined as reusable, and that are accessed in either LSR or NSR mode. CVDA values are:

EMPTYREQ

If the data set is defined as reusable, it is set to empty the next time a file that references it is opened in non-RLS mode.

Notes:

- If you specify `EMPTYREQ` for a nonreusable data set, CICS accepts it, but a subsequent attempt to open the file fails.
- If you specify `EMPTYREQ` for a file defined with `RLSACCESS(YES)`, CICS accepts it, but the option does not have any effect unless the file is subsequently opened in non-RLS mode.
- If you specify `EMPTYREQ` for a file that refers to a coupling facility data table that requires preloading from a data set and is specified with `RLSACCESS(NO)`, and opening the file triggers the table load, the data set is set to empty.
- If you specify `EMPTYREQ` for a file that refers to a coupling facility data table that does not require loading from a source data set, the option is ignored.
- If you specify `EMPTYREQ` for a file that refers to a coupling facility data table that is already loaded from a source data set, the option is ignored.

NOEMPTYREQ

The data set has been defined as reusable but is not set empty the next time a file that references it is opened. Specify `NOEMPTYREQ` for a coupling facility data table.

ENABLESTATUS(*cvda*)

Specifies whether application programs can access the file. CVDA values are as follows:

DISABLED

The file is unavailable for access by application programs.

ENABLED

The file is available for access by application programs.

EXCLUSIVE(*cvda*) (BDAM only)

Specifies whether records on this file are placed under exclusive control when a read for update is issued. CVDA values are as follows:

EXCTL

Records on this file are under exclusive control.

NOEXCTL

Records on this file are not under exclusive control.

FILE(*data-value*)

Specifies the 8-character file name as defined to CICS.

KEYLENGTH(*data-value*) (CFDT only)

Specifies, as a fullword binary value, the key length of records in a coupling facility data table. To set a key length, specify a value in the range 1 - 16. To clear a key length (set it to null values), specify `KEYLENGTH(0)`.

You can specify the key length for a file that does not currently refer to a coupling facility data table, but which could be switched to use a coupling facility data table at a later date.

LOADTYPE(*data-value*) (CFDT only)

Specifies whether the coupling facility data table associated with the file requires pre loading from a source data set. CVDA values are as follows:

LOAD

The coupling facility data table requires loading from a source data set

before it is fully usable; the transactions that use this coupling facility data table rely on it containing the records from the specified source data set.

NOLOAD

The coupling facility data table does not require loading from a source data set; it is fully usable as soon as it is created, and is populated by the transactions that use it.

You can specify the load type for a file that does not currently refer to a coupling facility data table, but which might be switched to use a coupling facility data table later.

LSRPOOLID(*data-value*) (**VSAM only**)

This attribute is obsolete, but is supported to provide compatibility with earlier releases of CICS.

A value specified for LSRPOOLID is transferred to the new option LSRPOOLNUM.

LSRPOOLNUM(*data-value*) (**VSAM only**)

Specifies, as a fullword binary value, the number of the LSR pool associated with this file. LSR pool IDs are in the range 1 through 255.

If the file cannot share buffers, set this value to 0.

For a CICS-maintained or user-maintained data table, the value must be 1 or greater. Both these types of CICS shared data tables must use LSR access mode (unless the file is defined to be opened in RLS access mode).

For a coupling facility data table, you can set this value to 0.

MAXNUMRECS(*data-value*)

Specifies, as a fullword binary value, the maximum number of records that the data table for this file can hold. Use this parameter to control the use of storage.

For any type of table, if you want to set a limit, specify a value in the range 1 - 99999999. If you do not want any limit to apply, specify MAXNUMRECS(0), which CICS interprets as no limit, and sets internally to the maximum positive fullword value (+2147483647 or X'7FFFFFFF').

To specify MAXNUMRECS for a recoverable coupling facility data table, use a value that is 5 - 10% more than the maximum number of records that the table is expected to contain. This value allows for additional records that might be created internally for processing recoverable requests. The margin to be left for this internal processing depends on the level of use of the coupling facility data table, and the nature of that use. An effect of this is that the NOSPACE condition (with a RESP2 value of 102) can be raised on a WRITE or REWRITE request to a recoverable coupling facility data table that has fewer records than the MAXNUMRECS limit specifies.

OPENSTATUS(*cvda*)

Specifies whether the file is open or closed. CVDA values are as follows:

CLOSED

The file is to be closed.

The close request is deferred until all units of work that hold repeatable read locks reach their sync points.

A coupling facility data table remains in existence (in the coupling facility) after the file is closed, unlike a user-maintained data table,

which ceases to exist when the file in the file-owning region is closed. Closing a file for a coupling facility data table does not prevent it being accessed through another file or by other CICS regions.

You can use the MVS MODIFY command to issue CFDT server commands:

- To set the table unavailable (MODIFY *server-name*,SET TABLE=*tablename*,AVAILABLE=NO) so that no other files can issue opens against it
- To delete the table from the coupling facility (MODIFY *server-name*,DELETE TABLE=*table-name*) if you do not want it to exist after the last file using it has been closed.

OPEN The file is opened.

For a coupling facility data table, open processing causes the coupling facility data table server to create the table if it does not exist when CICS processes the open request. If the installed file definition specifies the name of a source data set, the coupling facility data table is created by loading the data from the source data set.

If a SET FILE(*filename*) OPEN command refers to a file that specifies LOAD(YES), but which does not name the source data set, the CFDT can be created and loaded only by opening a file that defines the source data set name.

If a recoverable data set is closed, the task issuing the close must commit any prior changes to that data set; otherwise, the request is rejected by file control.

READ(*cvda*)

Specifies whether records can be read from the file. CVDA values are as follows:

NOTREADABLE

Records are not readable from the file.

READABLE

Records are readable from the file.

READINTEG(*cvda*)

Specifies the default level of read integrity for the file. CVDA values are as follows:

CONSISTENT

Consistent read integrity is required for this file.

REPEATABLE

Repeatable read integrity is required for this file.

UNCOMMITTED

No read integrity is required for this file.

These default read integrity values are used only when the file read request does not specify read integrity options explicitly on the EXEC CICS command.

CICS ignores a READINTEG option specified for a coupling facility data table.

RECORDSIZE(*data-area*) (CFDT only)

Specifies, as a fullword binary value, the maximum record size for a coupling facility data table in the range 1 - 32767.

You can specify the record size for a file that does not currently refer to a coupling facility data table, but which might be switched to use a coupling facility data table later. Specify a record size of zero to remove a previously defined value.

RLSACCESS(*cvda*)

Specifies whether the file is to be accessed in RLS mode. The file must be closed, and either disabled or unenabled, to change the access mode to RLS access or to non-RLS access.

The non-RLS mode becomes either LSR or NSR, depending on the value specified for LSRPOOLNUM in the file resource definition.

CVDA values are as follows:

NOTRLS

The file is opened in LSR or NSR mode when it is next opened.

RLS The file is opened in RLS access mode when it is next opened.

See the *CICS Recovery and Restart Guide* for information about switching between RLS and non-RLS modes.

STRINGS(*data-value*) (**VSAM only**)

Specifies, as a fullword binary value, the maximum number of concurrent operations to allow on this file, in the range 1 - 255.

TABLE(*cvda*) (**VSAM and CFDT only**)

Specifies whether the file name specified on the FILE parameter represents a data table. CVDA values are as follows:

CFTABLE

The file name refers to a coupling facility data table.

CICSTABLE

The file name represents a CICS-maintained data table.

NOTTABLE

The file name does not represent a data table.

USERTABLE

The file name represents a user-maintained data table.

TABlename(*data-area*) (**CFDT only**)

Specifies the 1- to 8-character name of the coupling facility data table to which this file refers. If TABlename is not specified, the table name defaults to the name of the file.

You can specify the table name for a file that does not currently refer to a coupling facility data table, but which might be switched to use a coupling facility data table later.

UPDATE(*cvda*)

Specifies whether the file is read-only or read/write. CVDA values are as follows:

NOTUPDATABLE

You can only read the records.

UPDATABLE

You can read, write, or delete the records.

UPDATEMODEL(*cvda*) (**CFDT only**)

Specifies the type of update coupling facility data table used for a coupling facility data table. CVDA values are as follows:

CONTENTION

The CFDT is to use the contention model, in which records are not locked when they are read for update, but an error is returned on a subsequent REWRITE or DELETE if the record has changed or been deleted since it was read for update.

LOCKING

The CFDT is to use the locking coupling facility data table, in which records are locked when they are read for update.

You can specify the update model for a file that does not currently refer to a coupling facility data table, but which might be switched to use a coupling facility data table later.

Conditions

FILENOTFOUND

RESP2 values:

- 18 The named file cannot be found.

INVREQ

RESP2 values:

- 1 The named file is REMOTE.
- 2 The named file is not CLOSED.
- 3 The named file is not DISABLED or UNENABLED.
- 4 ADD has an invalid CVDA value.
- 5 BROWSE has an invalid CVDA value.
- 6 BUSY has an invalid CVDA value.
- 7 DELETE has an invalid CVDA value.
- 8 DISPOSITION has an invalid CVDA value.
- 9 EMPTYSTATUS has an invalid CVDA value.
- 10 LSRPOOLNUM is specified for a non-VSAM data set.
- 11 LSRPOOLNUM is not in the range 1 - 255, or the corresponding buffer is not defined.
- 12 READ has an invalid CVDA value.
- 13 STRINGS value is not in the range 1 - 255, or this is not a VSAM file.
- 14 UPDATE has an invalid CVDA value.
- 16 OPENSTATUS has an invalid CVDA value.
- 17 ENABLESTATUS has an invalid CVDA value.
- 19 DELETE has been specified for a non-VSAM file.
- 20 EMPTYSTATUS has been specified for a non-VSAM file.
- 21 CLOSED or DISABLED has been specified by a task that has issued one or more recoverable requests within the current unit of work.
- 22 ENABLED was specified for a file that is currently DISABLING or UNENABLING.
- 23 EXCLUSIVE has an invalid CVDA value.
- 24 EXCLUSIVE has been specified for a non-BDAM file.
- 28 OPEN, CLOSE, ENABLE, or DISABLE has been specified but an exit program running at exit point XFCSREQ instructed CICS not to carry out the command.
- 29 TABLE has an invalid CVDA value.
- 30 MAXNUMRECS value is out of range.
- 31 The TABLE option is invalid for a BDAM file (must be VSAM for a data table).
- 32 The TABLE option is invalid for a file defined with the REUSE option.
- 33 The TABLE option is invalid for a file defined as UNBLOCKED.
- 34 The MAXNUMRECS option is invalid for a BDAM file (must be VSAM for a data table).

- 35 The MAXNUMRECS option is invalid for a file defined with the REUSE option.
- 36 The MAXNUMRECS option is invalid for a file defined as UNBLOCKED.
- 37 The TABLE option is invalid when LSRPOOL=0 is specified.
- 39 The USERTABLE option is invalid when record format is not variable.
- 40 CONSISTENT or REPEATABLE is specified for a file that is not accessed in RLS mode.
- 41 The DSNB cannot be disconnected, and a new DSNB cannot be connected, for this file because the file has deferred work outstanding, for which there are retained or repeatable read locks. In this case, at least one shunted UOW is awaiting completion, that has made changes to this file.
- 42 The SET FILE request cannot be satisfied because the file has deferred work outstanding, for which there are retained or repeatable read locks. In this case, at least one shunted UOW is awaiting completion, that has made changes to this file. The only valid options when a file has deferred work pending are those that change the file state. File state changes are permitted because they might be required to enable the deferred work to be completed.
- 43 The file cannot be discarded because it has deferred work outstanding, for which there are retained or repeatable read locks. In this case, at least one shunted UOW, awaiting completion, has made changes to this file.
- 44 A file open request cannot be satisfied because the file refers to a data set that is marked as unavailable by a SET DSNAME UNAVAILABLE command.
- 45 A file open request cannot be satisfied because the file references an RLS-mode data set that was quiesced by a SET DSNAME QUIESCED command.
- 46 A file open request cannot be satisfied because the file refers to an RLS-mode data set that is being copied by a DFSMSdss-initiated non-BWO backup.
- 47 A file open request cannot be satisfied because the file references an RLS-mode data set that is in the process of quiescing by a SET DSNAME QUIESCED command.
- 48 A file open request cannot be satisfied because the file refers to a data set for which its ICF Catalog entry indicates that a recovery is pending or is in progress; for example, a CICS VSAM Recovery job is running.
- 49 An invalid CVDA is specified for the READINTEG option.
- 50 An attempt has been made to open an RLS file but RLS is not supported, either because the level of VSAM does not support RLS or because RLS=NO has been specified during system initialization.
- 51 An invalid CVDA is specified for the RLSACCESS option.
- 52 An attempt has been made to specify RLS access for a BDAM data set.
- 53 An attempt has been made to specify a CICS-maintained data table for a file defined with RLS access.
- 54 A file open request cannot be satisfied because of one of the following reasons:
- The file is being opened in RLS mode and this region has other files open in non-RLS mode against the data set that it references.
 - The file is being opened in non-RLS mode and this region has other files open in RLS mode against the data set that it references.
 - The file is being opened in non-RLS mode and this region has unresolved RLS recovery work against the data set that it references.
- 55 LOADTYPE has an invalid CVDA value.

- 56 UPDATEMODEL has an invalid CVDA value.
- 57 EMPTYSTATUS has a CVDA value that is not allowed for a coupling facility data table. EMPTYSTATUS must be NOEMPTYREQ for a coupling facility data table.
- 58 CFDTPOOL is not specified for a file that refers to a coupling facility data table.
- 59 KEYLENGTH is not specified for a file that refers to a coupling facility data table, and which specifies LOAD=NO.
- 60 An invalid KEYLENGTH is specified. The KEYLENGTH must be in the range 1 - 16 for a coupling facility data table.
- 61 RECORDSIZE is not specified for a file that refers to a coupling facility data table that specifies LOAD=NO.
- 62 An invalid RECORDSIZE is specified. RECORDSIZE must be in the range 0 - 32 767 bytes.
- 63 OPEN is specified for a file that refers to a coupling facility data table, but OPEN processing has failed because of one of these reasons:
 - The file attributes do not match those specified when the coupling facility data table was created.
 - A keylength or recordsize has been specified that exceeds the maximum supported.
- 64 OPEN is specified for a file that refers to a coupling facility data table, but OPEN processing has failed because the server is unavailable.
- 65 An invalid CFDTPOOL name is specified.
- 66 An invalid TABLE name is specified.
- 67 An UPDATEMODEL of CONTENTION is specified for a recoverable coupling facility data table. The update model must be LOCKING for a coupling facility data table that is recoverable.
- 69 The DSNNAME is invalid.

IOERR

RESP2 values:

- 0 The command failed before the request was passed to the resource management system.
- * OPEN has failed outside file control. The RESP2 field contains the response that was returned to file control by the external resource management system.
- * CLOSE has failed outside file control. The RESP2 field contains the response that was returned to file control by the external resource management system.

In all cases of IOERR, examine the CICS console for messages that provide more information about the error. Resource management system refers to any of the catalog management systems such as VSAM, BDAM, or DFSMS, or to an external resource management system such as CFDT or SDT (Coupling facility data tables, or Shared data tables).

Depending on how you are viewing the RESP2 value, note that it might contain the decimal equivalent of the response returned to file control.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

Examples

```
EXEC CICS SET FILE ('FILE12')
          WAIT
          CLOSED
          DISABLED
          DELETABLE
          LSRPOOLNUM(7)
          STRINGS(50)
EXEC CICS SET FILE ('FILE12')
          OPEN
          ENABLED
```

On the first command, the WAIT option tells CICS to allow all activity on FILE12 to quiesce before closing the file, and to return control to the issuing application only when this request has been started. When the file is CLOSED, it is DISABLED. Delete commands are allowed, LSRPOOL number 7 is associated with the file, and up to 50 concurrent operations are allowed.

The second of the two commands opens and then enables the file. Setting a file CLOSED and DISABLED makes the file eligible for deletion (DISCARD) or reinstallation by another task. Thus, another task can delete the file after the first SET command but before the second SET command.

SET HOST

Sets the status of a virtual host to enabled or disabled.

SET HOST

```
►►—SET HOST(data-area)—┐└─ENABLESTATUS(cvda)—┘◄◄
```

Conditions: NOTAUTH, NOTFND

This command is threadsafe.

Description

The SET HOST command is used to set the status of a virtual host to enabled or disabled. Disabling a virtual host means that all the URIMAP definitions that make up the virtual host cannot be accessed by applications. When a virtual host is disabled, CICS returns a HTTP response with a 503 (Service Unavailable) status code to Web clients.

When the INQUIRE URIMAP command is used to inquire on an individual URIMAP definition, a special status DISABLEDHOST is returned to indicate that the virtual host is disabled. You do not need to change the disabled status of the URIMAP definitions individually; the SET HOST command can be used to re-enable all the URIMAP definitions that make up the virtual host. However, note that a URIMAP definition with the DISABLEDHOST status cannot be discarded. If you want to discard the definition, it must be disabled individually (using the SET URIMAP command).

Options

HOST(*data-area*)

specifies the name of a virtual host. The name of each virtual host is taken

from the host name specified in the URIMAP definitions that make up the virtual host. For example, if your CICS region contained URIMAP definitions that specified a host name of `www.example.com`, CICS would create a virtual host with the name `www.example.com`. A host name in a URIMAP definition can be up to 120 characters.

ENABLESTATUS (*cvda*)

CVDA values are:

ENABLED

The URIMAP definitions that make up the virtual host can be accessed by applications.

DISABLED

The URIMAP definitions that make up the virtual host cannot be accessed by applications.

Conditions

NOTAUTH

RESP2 values are:

100 The user associated with the issuing task is not authorized to use this command.

NOTFND

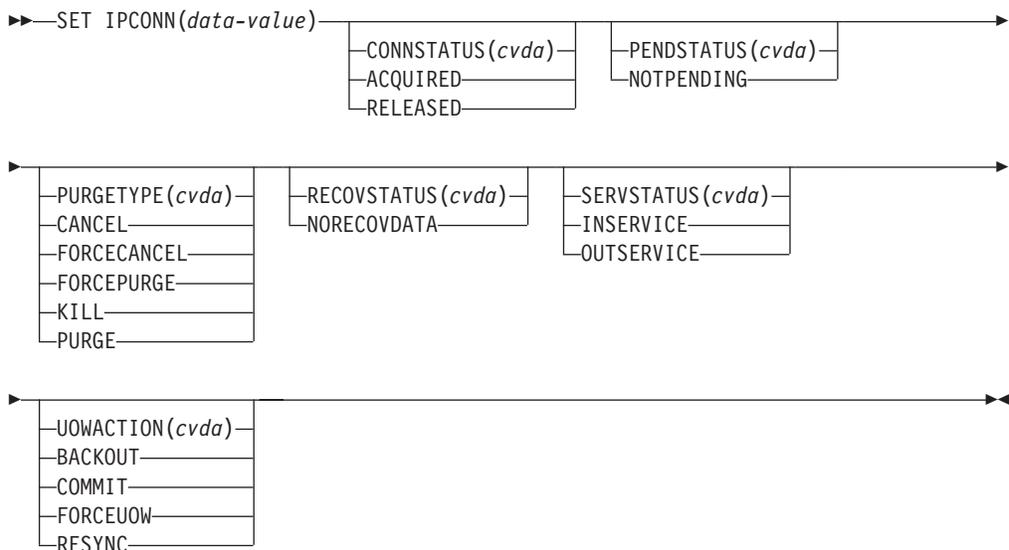
RESP2 values are:

5 The virtual host cannot be found.

SET IPCONN

Change the attributes of an IPCONN or cancel outstanding AIDs.

SET IPCONN



Conditions: INVREQ, IOERR, NORMAL, NOTAUTH, SYSIDERR

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

This command is threadsafe.

Description

The SET IPCONN command allows you to change some of the attributes that define an IPCONN. Control returns to the issuing program when the required operation has been scheduled. To get the operation started, it is necessary to relinquish control to CICS.

Note: SET IPCONN is used to change the attributes of IPIC connections (also known as "IPCONN"s"). See also "SET CONNECTION" on page 614. The SET CONNECTION command is used to change the attributes of MRO and ISC over SNA connections.

For information about the different kinds of intercommunication connections, see the *CICS Intercommunication Guide*.

Options

CONNSTATUS(*cvda*)

specifies whether to acquire or release sessions with the system represented by the IPCONN name. An IPCONN cannot be both ACQUIRED and OUTSERVICE.

CVDA values are:

ACQUIRED

Sessions are to be acquired.

RELEASED

Sessions are to be released.

For further information about managing IPCONNs, see the Intercommunication overview in the *Intercommunication Guide*.

IPCONN(*data-value*)

specifies, as an 8-character field, the name of the IPCONN to be modified. This is the name of the remote system or region specified on the IPCONN option of the IPCONN definition.

PENDSTATUS(*cvda*)

specifies, for an IPCONN to a CICS Transaction Server for z/OS partner that has performed an initial start, that the normal resynchronization process is to be overridden:

The CVDA value is:

NOTPENDING

Forces all indoubt units of work (that were created by the IPCONN before the initial start of the partner) to either commit or back out, as specified by the ACTION option of the TRANSACTION definition. It also forgets any resyncs (waitforget UOW-links) that are outstanding for the connection, and created before the initial start of the partner.

The PENDING condition indicates the existence of recovery information (either shunted UOWs or decisions remembered for the partner) on a connection that has experienced a lognames mismatch with its partner. This indicates that the partner has performed an initial start and that the recovery protocol has been corrupted by a loss of log data at the partner.

It is not possible to set a connection to NOTPENDING state (forcing indoubt and erasing NOFORGET UOWs) until CICS has made contact with the partner and received a new logname from it.

Decisions for a whole connection can be forgotten, but that does not affect the memory of a decision for any other connection involved in the UOW.

Note: SET IPCONN NOTPENDING, SET IPCONN NORECOVDATA, and SET IPCONN UOWACTION are mutually exclusive. For advice on which command to use, see the notes following the description of the UOWACTION option.

The exchange lognames function and the resynchronization function are described in the *CICS Intercommunication Guide*.

PURGETYPE (*cvda*)

specifies how associated transactions are to be purged. CVDA values are:

CANCEL

specifies that queued requests by transactions to use this IPCONN are to be canceled.

Queued requests to use this IPCONN by CICS system transactions that manage communications across this IPCONN are not purged unless FORCECANCEL is specified.

Message DFHISnnnn is written to CSMT to indicate how many queued requests to use this IPCONN have been deleted and how many remain.

A "QUEUED REQUESTS CANCELED" message appears on the CEMT panel whenever queued requests to use this IPCONN are deleted using the CANCEL option of the CEMT SET IPCONN command.

FORCECANCEL

specifies that all queued requests by transactions to use this IPCONN are to be canceled, including requests by CICS system transactions that manage communications across this IPCONN. This can lead to unpredictable results and should be used only in exceptional circumstances.

A "QUEUED REQUESTS CANCELED" message appears on the CEMT panel whenever queued requests to use this IPCONN are deleted using the FORCECANCEL option of the CEMT SET IPCONN command.

FORCEPURGE

specifies that all transactions running on sessions to the connected system are to be abnormally terminated immediately. This can lead to unpredictable results and should be used only in exceptional circumstances.

In some extreme cases (for example, if an error occurs during backout processing), CICS might terminate abnormally.

KILL specifies that the task is to be terminated. System and data integrity is not guaranteed. The KILL option extends the PURGE and FORCEPURGE options. You should use it only after an attempt has been made to PURGE or FORCEPURGE a task. The KILL option does not guarantee integrity of any kind but in some situations it allows you to free up a stalled region, enabling the region to continue

processing. In some cases, for example if a task is killed during backout processing, CICS terminates abnormally.

PURGE

specifies that transactions running on the connected system are to be abnormally terminated. Transactions are terminated only if system and data integrity can be maintained. A transaction is not purged if its definition specifies SPURGE=NO.

RECOVSTATUS(*cvda*)

specifies that the normal resynchronization process is to be overridden. The CVDA value is:

NORECOVDATA

Forces all indoubt units of work (according to the transaction definitions), targets any resyncs that were outstanding for the IPCONN, and erases the logname previously received from the partner system. The state of the connection is reset.

Attention: You should use SET IPCONN NORECOVDATA only in exceptional circumstances. It erases recovery information and may compromise data integrity for units of work that have updated resources on remote systems.

Examples of circumstances in which you might need to use it are:

- You need to discard an IPCONN, and it is not possible for the quiesce protocols with the partner system to be completed.
- An operational or logic error results in a logname mismatch for the connection. The connection state must be reset to allow the exchange lognames process to complete.

Note: SET IPCONN NORECOVDATA, SET IPCONN NOTPENDING, and SET IPCONN UOWACTION are mutually exclusive.

SERVSTATUS(*cvda*)

specifies whether the IPCONN is to be placed in service or out of service. CVDA values are:

INSERVICE

The IPCONN is to be placed in service. This allows it to be acquired.

OUTSERVICE

The IPCONN is to be placed out of service; that is, not available for use.

The following occurs:

- If the connection is currently ACQUIRED and you specify OUTSERVICE, the command fails with INVREQ and a RESP2 of 2. You must RELEASE the connection before setting OUTSERVICE.
- If the connection is currently RELEASED, the status of the connection is set OUTSERVICE and it cannot be used until it is INSERVICE again.

UOWACTION(*cvda*)

specifies that the normal resynchronization process is to be partially overridden: decisions are taken for any units of work that are indoubt because of a failure of the IPCONN; but the decisions are recorded and any data inconsistencies are reported when the connection is next acquired.

The operation is synchronous with setting the state of the UOW; that is, an INQUIRE UOW following a SET IPCONN UOWACTION returns the new UOW states. CVDA values are:

BACKOUT

All UOWs shunted because of the failure of this IPCONN are to be backed out.

COMMIT

All UOWs shunted because of the failure of this IPCONN are to be committed.

FORCE

All UOWs shunted because of the failure of this IPCONN are to be forced to BACKOUT or COMMIT, as specified on the ACTION option of the TRANSACTION definition.

RESYNC

Any UOWs shunted because of the failure of this IPCONN are to be retried (that is, exchange lognames resynchronization for this connection is to be attempted). This process should normally be started automatically when a connection is acquired or when a UOW is unshunted.

Notes:

1. SET IPCONN UOWACTION unshunts all units of work that have failed indoubt because of a failure of the IPCONN. Before issuing SET IPCONN FORCE, you may want to use the SET UOW command to specify commit or backout for each indoubt unit of work explicitly, rather than letting it default. Local procedures will determine the importance of the data and the method of using the INQUIRE UOW, INQUIRE UOWENQ, and INQUIRE UOWLINK commands to establish the correct actions.
2. As far as shunted units of work are concerned, you may use only one of SET IPCONN UOWACTION, SET IPCONN NOTPENDING, and SET IPCONN NORECOVDATA. SET IPCONN NORECOVDATA should be used only in exceptional circumstances.
3. To force all indoubt units of work caused by a failure of the IPCONN in the same direction, use SET IPCONN COMMIT or SET IPCONN BACKOUT.
4. Neither SET IPCONN UOWACTION nor the SET UOW UOWACTION command clears resync information. If you want to do this, you must use SET IPCONN NOTPENDING or SET IPCONN NORECOVDATA.
5. You can issue SET UOW UOWACTION commands *before* issuing SET IPCONN NOTPENDING or SET IPCONN NORECOVDATA.

Conditions

INVREQ

RESP2 values:

- 2 ACQUIRED and OUTSERVICE are specified inconsistently in any of the following ways:
 1. ACQUIRED specified with OUTSERVICE
 2. ACQUIRED specified for OUTSERVICE IPCONN resource
 3. RELEASED and OUTSERVICE specified in the same command for an ACQUIRED IPCONN resource.
 4. OUTSERVICE specified for an IPCONN resource that is not RELEASED

- 3 CONNSTATUS has an invalid CVDA value.
- 4 SERVSTATUS has an invalid CVDA value.
- 7 PURGETYPE has an invalid CVDA value.
- 8 PENDSTATUS has an invalid CVDA value.
- 18 NOTPENDING cannot be set for an IPCONN that has successfully completed exchange lognames processing.
- 19 CONNSTATUS cannot be set to ACQUIRED when in the FREEING state.
- 20 An attempt was made to acquire a one-way IPCONN.
- 21 BACKOUT or FORCE was specified, but was unsuccessful. Some UOWs remain shunted for this IPCONN.
- 22 Other SET parameters were included with the CANCEL or FORCECANCEL option.
- 25 IPCONN is still in service.
- 26 RECOVSTATUS does not have a value of NORECOVDATA.
- 27 The CVDA value specified on the UOWACTION option is invalid.
- 45 NORECOVDATA cannot be set for an IPCONN that is in service.

IOERR

RESP2 values:

- 10 Unexpected error.

NORMAL

RESP2 values:

- 58 AIDs have been successfully canceled.
- 59 No AIDs have been canceled.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SYSIDERR

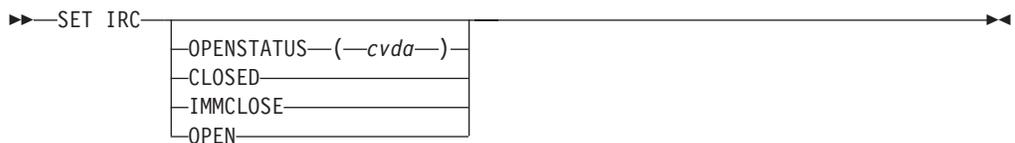
RESP2 values:

- 9 The named IPCONN could not be found.

SET IRC

Open or close interregion communication.

SET IRC



Conditions: INVREQ, IOERR, NOSTG, NOTAUTH

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.

Description

The SET IRC command allows you to start (open) or stop (close) interregion communication (IRC) in your CICS region. IRC must be open for your region to communicate with another CICS region using a multiregion operation (MRO) connection, or for a non-CICS client region to use your CICS over an external CICS interface (EXCI) connection.

Support for this type of communication must be specified at CICS startup (in the ISC initialization option), and at least one CONNECTION resource must be defined with an ACCESSMETHOD value indicating MRO; otherwise exception conditions occur when you attempt to open IRC. The *CICS Intercommunication Guide* describes the various requirements.

Options

OPENSTATUS(*cvda*)

specifies whether IRC communications should be started (open) or stopped (closed), and if CICS needs to stop IRC, whether tasks using MRO should be allowed to complete first. CVDA values are:

CLOSED

IRC is to be stopped. If it is currently open, CICS is to quiesce all MRO activity and then close IRC. Tasks using CICS-to-CICS MRO sessions and EXCI sessions are allowed to complete before closure, but new tasks requiring IRC are not begun.

IMMCLOSE

IRC is to be stopped. If currently open, CICS is to terminate abnormally any tasks using IRC immediately and then close IRC.

OPEN IRC is to be started. If currently closed, CICS is to open it.

Conditions

INVREQ

RESP2 values:

- 1 A program required for IRC, DFHCRSP, is unavailable.
- 2 OPENSTATUS has an invalid CVDA value.
- 4 CICS was initialized without IRC support (ISC=NO).
- 5 No connection has been defined.
- 6 The z/OS Communications Server APPLID for this CICS is blanks; IRC requires a non-blank APPLID.
- 7 Another CICS using IRC has the same z/OS Communications Server APPLID as this one; unique names are required.
- 8 IRC rejected the open of this CICS because it had already reached the maximum number of logons.
- 18 IRC support (the DFHIRP module) is below the level required by this CICS system.

IOERR

RESP2 values:

- 12 IRC initialization failed.
- 13 The log on to IRC failed.

- 14 An attempt to attach the node error transaction, CSNC, failed.
- 15 An error occurred closing IRC.

NOSTG

RESP2 values:

- 9 CICS storage is insufficient for the request.
- 10 MVS storage is insufficient (SVC block request rejected).
- 11 MVS storage is insufficient (SUBSYS block request rejected).

NOTAUTH

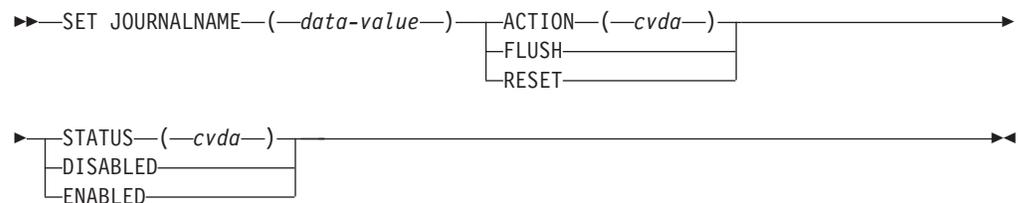
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SET JOURNALNAME

Enable or disable a CICS user journal.

SET JOURNALNAME



Conditions: INVREQ, IOERR, JIDERR, NOTAUTH

Description

The SET JOURNALNAME command allows you to enable or disable a CICS user journal.

SET JOURNALNAME has no effect on a journal that is being used as the forward recovery log or autojournal for a VSAM file until the next time the file is opened. It has no effect on the system log.

You can use SET JOURNALNAME for a journal name that is not currently known to CICS. CICS dynamically creates an entry for the specified journal and, if necessary, defines it to the MVS system logger using a matching JOURNALMODEL definition.

The ability to issue SET JOURNALNAME commands for journal names not known to CICS enables you to perform log stream connection processing before the corresponding journals are first referenced. For example, you could do this during a PLT program at initialization to avoid the delay that normally occurs at first reference.

Options

ACTION (*cvda*)

specifies the action you want CICS to take for the specified journal name. CVDA values are:

FLUSH

The log buffers are written out to the log stream, but the journal is not closed.

You can use this option to ensure that all current records are written out to the log stream before processing the stream using a batch utility.

In the case of autojournals and forward recovery logs, the FLUSH is forced if the file is open (the FLUSH does not wait until the next time the file is opened).

RESET

The journal is disconnected from its log stream, but can be reopened by a journal write.

Note: ACTION and STATUS are mutually exclusive options. If you specify ACTION, you cannot also specify STATUS.

JOURNALNAME (*data-value*)

specifies the name of the journal.

To modify journals defined with a numeric identifier in the range 1–99, specify journal name DFHJ*nn*, where *nn* is the journal number.

You cannot specify DFHLOG or DFHSHUNT, because you are not allowed to modify the status of the system log.

STATUS (*cvda*)

specifies the new status for the journal. The CVDA values are:

DISABLED

The journal is flushed then disabled. It cannot be used again until it is reenabled by the STATUS(ENABLED) or ACTION(RESET) options on a SET JOURNALNAME command.

ENABLED

The journal is open and is available for use.

Note: STATUS and ACTION are mutually exclusive options. If you specify STATUS, you cannot also specify ACTION.

Conditions

INVREQ

RESP2 values:

- 2 The request is invalid.
- 3 The system log cannot be changed.
- 4 The ACTION option has an invalid CVDA value.
- 5 The STATUS option has an invalid CVDA value.
- 7 The ACTION option specifies FLUSH or RESET for a journal that is not currently connected to a log stream.

IOERR

RESP2 values:

- 6 The log stream associated with the journal name cannot be connected to, or the journal cannot be opened, or an unrecoverable error has occurred during the flushing of the log buffer to the log stream.

JIDERR

RESP2 values:

- 1 The specified journal name was not found.
- 2 An error occurred during an attempt to define the log stream associated with the journal name, or the journal name has been incorrectly specified.
- 3 The specified journal name refers to a DASD-only log stream to which a CICS region in another MVS image is currently connected.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

SET JOURNALNUM

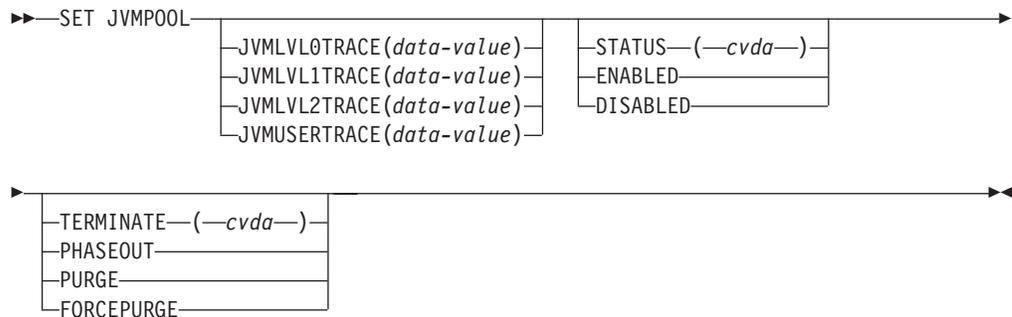
This command is replaced by the SET JOURNALNAME command for all supported releases for changing the OPENSTATUS setting of a journal.

All the options on the SET JOURNALNUM are obsolete, and the only runtime support provided by CICS for compatibility with earlier releases is to return the JIDERR exception condition. The translator translates the command, but issues a warning message.

SET JVMPOOL

Enable or disable the JVM pool and set JVM tracing options.

SET JVMPOOL



Conditions: INVREQ, NOTAUTH

This command is threadsafe.

Description

You can use the **SET JVMPOOL** command to enable or disable the JVM pool and change JVM tracing options. There is only one pool of JVMs in a CICS region, therefore no name or identifier is required on this command.

The **TERMINATE** option on this command is deprecated. Use the **TERMINATE** option on the **PERFORM JVMPOOL** command instead.

Options

JVMLVL0TRACE(*data-value*), **JVMLVL1TRACE**(*data-value*), **JVMLVL2TRACE**(*data-value*), **JVMUSERTRACE**(*data-value*)

Specify a 240-character setting for the JVM tracing options, which are used when the associated level of tracing has been activated using the CICS-supplied transaction CETR, the **EXEC CICS SET TRACETYPE** command, or the **SPCTRSJ** system initialization parameter. Trace levels 29–32 for the SJ component correspond to **JVMLVL0TRACE**, **JVMLVL1TRACE**, **JVMLVL2TRACE**, and **JVMUSERTRACE** respectively.

STATUS(*cvda*)

Specifies whether new Java requests can be accepted and serviced by the JVM pool. The CVDA values are as follows:

ENABLED

The pool status is set to enabled for use and Java programs can execute using pooled JVMs.

DISABLED

The pool status is set to disabled, preventing new requests from being serviced from the pool. Programs that were started before the command was issued can run to completion.

TERMINATE(*cvda*)

Specifies that all of the JVMs in the JVM pool are to be terminated.

The **TERMINATE** option on this command is deprecated. Use the **TERMINATE** option of the **PERFORM JVMPOOL** command instead. The **PERFORM JVMPOOL** command enables you to terminate a subset of the JVMs in the pool by JVM profile, which can reduce application downtime.

Conditions

INVREQ

RESP2 values:

- 1 TERMINATE is specified with an invalid CVDA value.
- 2 STATUS is specified with an invalid CVDA value.
- 3 A task purge failed.

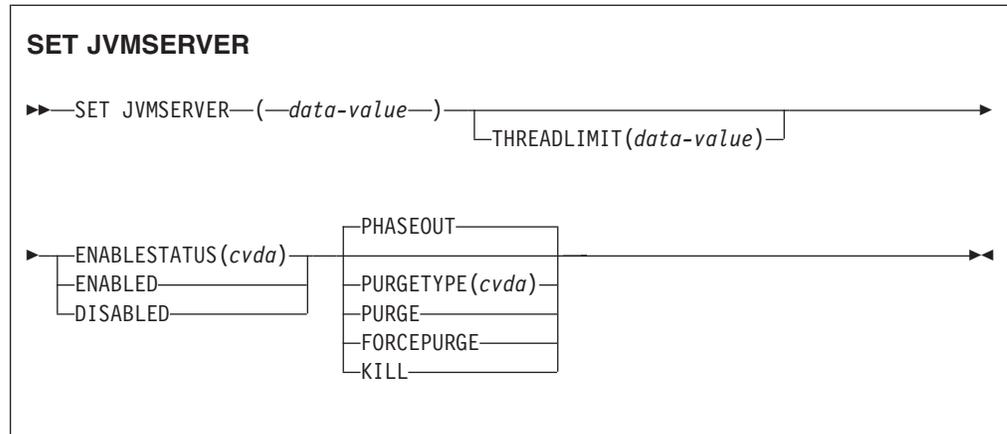
NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SET JVMSERVER

Change the status of an installed JVMSERVER resource.



Conditions: INVREQ, NORMAL, NOTAUTH, NOTFND

This command is threadsafe.

Description

The **SET JVMSEVER** command enables, disables, and modifies a JVM server.

Options

ENABLESTATUS(*cvda*)

Set the status of the JVMSEVER resource:

ENABLED

Enable the JVMSEVER resource. CICS creates a Language Environment enclave and starts a JVM.

DISABLED

Disable the JVMSEVER resource. CICS finishes processing any requests and then stops the JVM and the Language Environment enclave.

JVMSEVER(*data-value*)

Specify the 8-character name of the JVMSEVER resource that you want to change.

PURGETYPE(*cvda*)

Specifies how tasks associated with the named JVM server are to be purged when you disable the resource. If you do not set a value, CICS uses the PHASEOUT option.

FORCEPURGE

Tasks that are running in the JVM server are force purged by the **SET TASK FORCEPURGE** command. Any threads running in the JVM are stopped. If the JVMSEVER resource remains in the BEING DISABLED state, some tasks could not be force purged.

Use this option only after you have used the PURGE option. Data integrity is not guaranteed.

KILL

Tasks that are running in the JVM server are terminated by the **SET TASK KILL** command. Any threads running in the JVM are stopped. The JVMSEVER resource enters the DISABLED state and all work is terminated. However, CICS might be left in an unstable state.

Use this option only after you have used the PURGE and FORCEPURGE options. System and data integrity are not guaranteed. CICS might terminate abnormally.

PHASEOUT

Tasks that are running in the JVM server continue until completion, but no new work is accepted by the JVM server. When all the tasks are finished, the JVMSERVER resource enters the DISABLED state. PHASEOUT is the default value.

PURGE

Tasks that are running in the JVM server are purged by the **SET TASK PURGE** command. Any threads running in the JVM are stopped. CICS purges tasks only when system and data integrity can be maintained. If the JVMSERVER resource remains in the BEING DISABLED state, some tasks could not be purged.

THREADLIMIT (*data-value*)

Set the maximum number of threads that are allowed by CICS when calling the JVM. Each thread is attached using a T8 TCB. The valid range is 1 - 256 threads. If the thread limit exceeds the maximum of 1024 threads allowed for the CICS region, taking into account the other enabled and disabled JVMSERVER resources, CICS adjusts the value accordingly.

Conditions

INVREQ

RESP2 values:

- 1 Insufficient threads available to satisfy the requested maximum number.
- 2 ENABLESTATUS value is not valid.
- 3 THREADLIMIT value specified is not valid because it is 0 or greater than 256.
- 4 The Language Environment enclave was not created successfully.
- 7 The JVMSERVER cannot be disabled because it is in the enabling state.
- 8 You must PURGE the tasks associated with the JVM server before using the FORCEPURGE option.
- 9 The PURGETYPE option has an invalid CVDA value.
- 10 You must FORCEPURGE the tasks associated with the JVM server before using the KILL option.

NORMAL

RESP2 value:

- 1 The number of available threads is less than the THREADLIMIT value requested.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

NOTFND

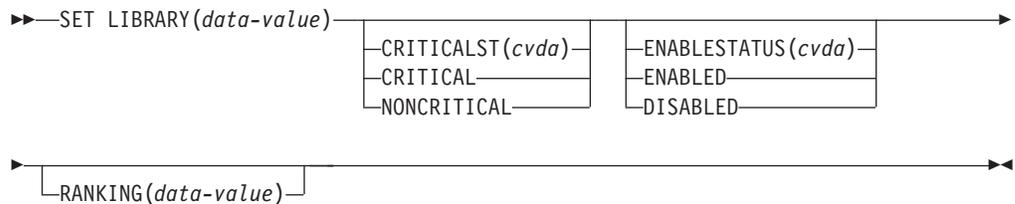
RESP2 value:

3 The JVMSERVER resource was not found.

SET LIBRARY

Change the attributes of a LIBRARY resource.

SET LIBRARY



Conditions: INVREQ, NOTAUTH, NOTFIND

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The SET LIBRARY command allows you to change some of the attributes of a particular LIBRARY resource installed in your CICS system.

Options

CRITICALST(*cvda*)

specifies whether the LIBRARY is critical to the start up of CICS. Values are set for the next warm or emergency restart only, not for the next cold or initial start. CVDA values are:

CRITICAL

The LIBRARY is critical to CICS startup. If the LIBRARY cannot be successfully installed during CICS startup for any reason, then a GO or CANCEL message is issued. The operator can decide whether to override the criticality and allow CICS to start or not. If CICS is allowed to continue, the LIBRARY is installed in a DISABLED status, unless install was not possible at all; for example, due to a short-on-storage condition.

If the reply is to continue with the startup, the LIBRARY will not be recatalogued as NONCRITICAL, so the critical status should be explicitly set to NONCRITICAL if it is decided that the LIBRARY should not be regarded as CRITICAL in future

NONCRITICAL

The LIBRARY is not critical to CICS startup. If the LIBRARY cannot be successfully installed during CICS startup, then the LIBRARY will be left in an installed but disabled state and a warning message will be issued, but CICS startup will continue.

LIBRARY(*data-value*)

specifies the 8-character name of the LIBRARY whose attributes are being changed

ENABLESTATUS(*cvda*)

specifies whether the LIBRARY will be included in the overall LIBRARY search order. CVDA values are:

DISABLED

The LIBRARY will not be included in the LIBRARY search order. The data sets in this LIBRARY concatenation will not be searched for program artifacts to load. Setting a LIBRARY to DISABLED will cause CICS to close the LIBRARY concatenation and to deconcatenate and unallocate the data sets in the LIBRARY

ENABLED

The LIBRARY will be included in the LIBRARY search order. The data sets in this LIBRARY concatenation will be searched for program artifacts to load.

Note: When a LIBRARY is disabled the information about where a program was loaded from becomes invalid, so when the LIBRARY is re-enabled a NEWCOPY or PHASEIN has to be issued before the program can be loaded again.

RANKING(*data-value*)

A fullword binary value containing a decimal number between 1 and 99 which specifies where this LIBRARY should appear in the overall LIBRARY search order relative to other LIBRARY concatenations. A lower number indicates that this LIBRARY will be searched for programs to load before other LIBRARY resources with higher ranking numbers. DFHRPL has a reserved ranking value of 10, and this ranking value cannot be specified for a dynamic LIBRARY.

LIBRARYs will appear in the search order, in order of ranking. LIBRARYs of equal RANKING will appear in the search order in the order in which they were installed or created in the local CICS system, with a LIBRARY that was installed earlier appearing before one that was installed later.

Conditions**INVREQ**

RESP2 values:

- 2 ENABLESTATUS has an invalid CVDA value.
- 3 CRITICALST has an invalid CVDA value.
- 4 RANKING value is out of range (less than 1 or greater than 99).
- 5 RANKING value is the reserved value of 10.
- 6 SET operations are not permitted for the static LIBRARY concatenation, DFHRPL.
- 7 The LIBRARY has not been enabled because one of the following occurred:
 - The LIBRARY failed to open.
 - Allocation of one or more of the data sets in the LIBRARY failed.
 - The LIBRARY concatenation failed.
- 7 A failure was encountered while attempting to deconcatenate the datasets during a LIBRARY disable operation.
- 8 A failure occurred while attempting to deallocate one or more of the data sets in the LIBRARY during a disable operation.
- 10 A failure occurred while attempting to close the LIBRARY during a disable operation.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFIND

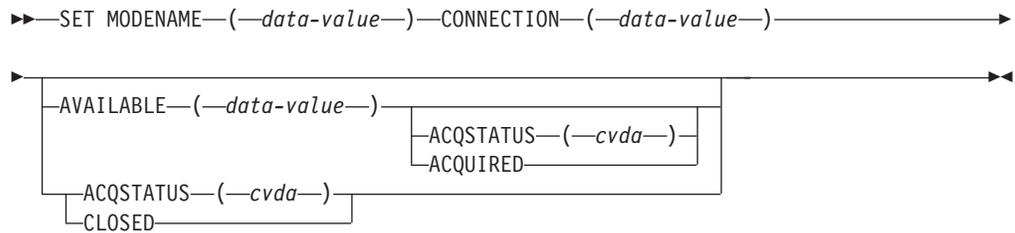
RESP2 values:

- 1 The named LIBRARY cannot be found.

SET MODENAME

Change the number of sessions in an APPC session group.

SET MODENAME



Conditions: INVREQ, NOTAUTH, SYSIDERR

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The **SET MODENAME** command enables you to increase or decrease the number of sessions **available** (bound) in a session group on a particular APPC connection. You identify the group to be changed by the MODENAME and CONNECTION values in its SESSIONS definition, rather than the name of the SESSIONS definition. You need both values because MODENAMES are not necessarily unique across connections.

SET MODENAME applies only to parallel session groups on an APPC connection on which CICS is already in session with its partner system, and only to groups created with a SESSIONS resource definition (not to SNASVCMG LU services manager sessions). The changes last only until the connection is released or the number of sessions is changed again.

If you increase the number of sessions, you can specify whether or not CICS should acquire the additional sessions; if you decrease the number, CICS unbinds the excess sessions automatically. If more than the target number of sessions are in use at the time of the command, CICS allows activity to quiesce before unbinding. Tasks using a session on the connection are allowed to complete, but new tasks requiring a session are not started until activity drops below the new limit.

Note: CICS uses a task that executes LU Services Manager transaction CLS1 to acquire or release sessions on parallel-session APPC connections. Data is passed to the task in a temporary storage queue whose name begins with the default prefix of DF. If your system defines queues named starting with DF as recoverable, CICS cannot initiate this task until a subsequent commit on the part of the task that issued the **SET MODENAME** command (either a **SYNCPOINT** command or an implicit syncpoint).

Options

ACQSTATUS(*cvda*)

specifies either that additional sessions are to be acquired if the AVAILABLE value increases the number, or that the number of available sessions is to be set to zero. CVDA values are:

ACQUIRED

Additional sessions, if any, are to be acquired.

CLOSED

The number of sessions is to be set to zero. CLOSED is equivalent to specifying AVAILABLE (0) and should not be specified with AVAILABLE. This value prevents either of the connected systems from using a session in the group.

AVAILABLE(*data-value*)

specifies, as a halfword binary value, the number of sessions to be available for use at any one time. The range for this value is from zero to the MAXIMUM value specified in the SESSIONS definition; you can determine this limit, if necessary, with an INQUIRE MODENAME command.

CONNECTION(*data-value*)

specifies the 4-character name of the connection for which this group of sessions is defined (from the CONNECTION value value in the SESSIONS definition).

MODENAME(*data-value*)

specifies the 8-character MODENAME value of the group of sessions that you are modifying (from its SESSIONS definition).

Conditions

INVREQ

RESP2 values:

- 3 MODENAME 'SNASVCMG' was specified.
- 4 The AVAILABLE value is out of range.
- 5 AVAILABLE was specified but CICS is not in session on this connection.
- 6 CLOSED was specified with AVAILABLE.
- 7 ACQSTATUS has an invalid CVDA value.
- 8 This is not a parallel-session APPC group.
- 9 ACQUIRED was specified but CICS is not in session on this connection.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SYSIDERR

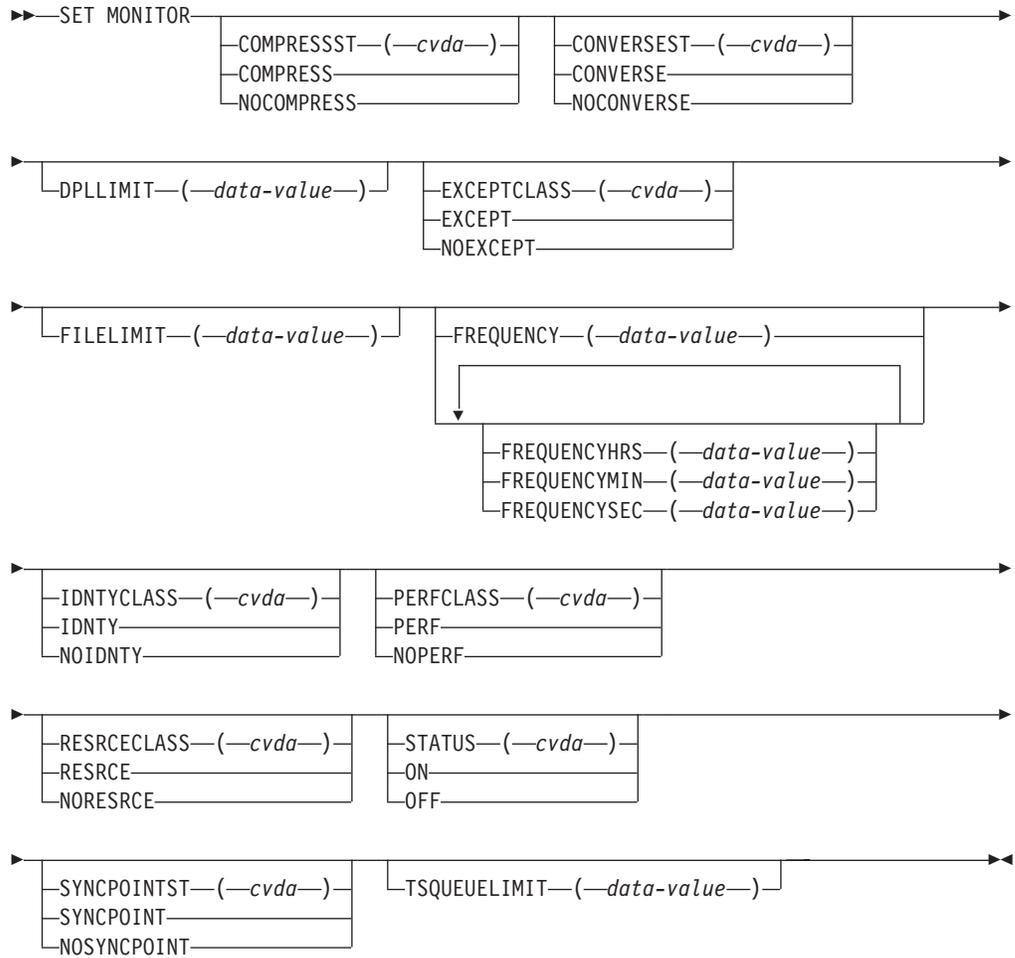
RESP2 values:

- 1 The connection cannot be found.
- 2 The MODENAME within the connection cannot be found.

SET MONITOR

Change CICS monitoring options.

SET MONITOR



Conditions: INVREQ, NOTAUTH

For more information about the use of CVDAs, see CICS-value data areas (CVDAs) in CICS System Programming Reference.

Description

Use the SET MONITOR command to switch CICS monitoring on or off, to modify the settings of the monitoring options, and to select the classes of monitoring data to be recorded.

CICS monitoring is controlled by a master switch (the STATUS option). Monitoring data is accumulated only while the STATUS option has the value ON and only for tasks that begin while STATUS is ON.

When monitoring is active, CICS accumulates the following types of data for each individual task:

- Performance data (types and counts of CICS commands and timings, for example)
- Exception data (waiting for a VSAM string, for example)
- Transaction resource data (counts and timings of the various file get, put, browse, add, and delete accesses, plus totals)
- Identity class data (counts, timings, and identifiers for transactions that have identity propagation data)

Additional switches determine which of these classes of monitor data are written to the SMF data set. Exception data is written only if EXCEPTCLASS is EXCEPT; transaction resource data is written only if RESRCECLASS is RESRCE; identity class data is written only if IDNTYCLASS is IDNTY; and performance data only if PERFCLASS is PERF. For an individual task, class data is recorded only if the class switch is on both at the time the task starts and at the time that class of data is written out.

Exception class data is written at the end of the event to which the exception applies. Performance class data is written at these specific times:

- At end of task
- At a terminal-receive wait, if the CONVERSEST value is CONVERSE
- At a frequency interval, if the interval is not zero
- At a sync point, if the SYNCPOINTST value is SYNCPOINT
- At a user event monitoring point with the DELIVER option

Identity class and transaction resource class data is written at the end of task only.

If you change STATUS from ON to OFF, CICS stops accumulating and recording monitoring data. Data for tasks in flight that is not already recorded is lost even if you turn monitoring back on before end of task.

Furthermore, if you are recording performance data, specify NOPERF in any command that sets monitoring OFF, to ensure that buffers containing recorded data for completed tasks are flushed; otherwise, some of this data can be lost.

If you leave STATUS on but turn one of the recording options off and then back on during a task, however, data loss depends on the class, as follows:

- Exception data is not written out for exceptions that occur while EXCEPTCLASS is NOEXCEPT but, if you change back to EXCEPT, subsequent exceptions are recorded.
- If you change PERFCLASS from PERF to NOPERF while a task is running, performance data already accumulated is recorded, but then recording stops. Accumulation continues, however. Therefore, if you change back to PERF before task end, no data is lost unless a monitor point with the DELIVER option occurs while NOPERF is in force. (DELIVER resets the counters.) The other conditions that ordinarily cause writing, sync point with a SYNCPOINTST value of SYNCPOINT, terminal receive wait with a CONVERSEST value of CONVERSE, or expiration of the frequency interval, do not reset the counts while recording is off, so that no counts are lost, although they might be combined.
- Transaction resource class and identity class data is written at the end of a task, and will be written only if the monitoring class is set (to RESRCE for the transaction resource class, or IDNTY for the identity class) at the point when the task ends.

Options

COMPRESSST (*cvda*)

Specifies whether you want data compression to be performed for the CICS SMF 110 monitoring records produced by the CICS monitoring facility. If you change the setting for the data compression option, the new setting applies to all monitoring records written from that point on, even if they are for a task being processed at the time the change is made. The new setting also applies to any records that are in the buffer waiting to be written to SMF at the time the change is made. The change applies only until a CICS restart.

COMPRESS

CICS is to perform data compression for the monitoring records. In some situations, some of the records might not be compressed. Data compression is the default.

NOCOMPRESS

CICS is not to perform data compression for the monitoring records.

CONVERSEST (*cvda*)

Specifies how CICS is to record performance data for conversational tasks (tasks that wait for terminal or session input).

CONVERSE

CICS is to produce a performance class record each time the task waits for terminal input as well as at task end, representing the part of the task since the previous wait or task start. Waits occur during execution of a CONVERSE command or a RECEIVE command that follows a SEND.

NOCONVERSE

CICS is to accumulate performance data across terminal waits and produce a single performance class record.

DPLLIMIT (*data-value*)

Specifies the maximum number of distributed program link requests for which CICS is to perform transaction resource monitoring, as a halfword binary value. The value specified must be in the range 0 - 64.

EXCEPTCLASS (*cvda*)

Specifies whether the exception class of monitoring data is to be recorded when monitoring is active. CVDA values are as follows:

EXCEPT

Exception data is to be recorded.

NOEXCEPT

Exception data is not to be recorded.

FILELIMIT (*data-value*)

Specifies the maximum number of files for which CICS is to perform transaction resource monitoring, as a halfword binary value. The value specified must be in the range 0 - 64.

FREQUENCY (*data-value*)

Specifies the interval at which CICS is to produce performance class records for long-running tasks. If a task runs longer than the frequency interval, CICS records its performance data separately for each interval or fraction. CICS can produce a performance class monitoring record in this way only when the long-running transaction is running on the QR or CO TCBs.

The frequency interval can be expressed in several ways:

- A 4-byte packed decimal composite, in the format 0hhmss+, using the FREQUENCY option.
- With separate hours, minutes, and seconds, using the FREQUENCYHRS, FREQUENCYMIN, and FREQUENCYSEC options. You can use these options singly or in any combination.

Whichever method you use, the interval value must be either zero or in the range from 1 minute to 24 hours. Zero specifies that CICS is to produce performance records only at task end, regardless of the length of the task.

In addition, if you use FREQUENCY or more than one of the separate options, the minutes and seconds portions of the value must not be greater than 59. The FREQUENCYMIN or FREQUENCYSEC options used alone can exceed 59. For example, you can express an interval of 1 hour and 30 minutes in any of the following ways:

- FREQUENCY(13000)
- FREQUENCYHRS(1), FREQUENCYMIN(30)
- FREQUENCYMIN(90)
- FREQUENCYSEC(5400)

FREQUENCYHRS (*data-value*)

Specifies the hours component of the frequency interval, in fullword binary form. See the FREQUENCY option.

FREQUENCYMIN (*data-value*)

Specifies the minutes component of the frequency interval, in fullword binary form. See the FREQUENCY option.

FREQUENCYSEC (*data-value*)

Specifies the seconds component of the frequency interval, in fullword binary form. See the FREQUENCY option.

IDNTYCLASS (*cvda*)

Specifies whether the identity class of monitoring data is to be recorded when monitoring is active. CVDA values are as follows:

IDNTY

Identity data is to be recorded.

NOIDNTY

Identity data is not to be recorded.

PERFCLASS (*cvda*)

Specifies whether the performance class of monitoring data is to be recorded when monitoring is active. CVDA values are as follows:

NOPERF

Performance data is not to be recorded.

PERF

Performance data is to be recorded.

RESRCECLASS (*cvda*)

Specifies whether CICS transaction resource monitoring is to be active or inactive. CVDA values are as follows:

NORESACE

CICS is not to perform transaction resource monitoring. No transaction resource data is accumulated or written to SMF.

RESRCE

CICS is to perform transaction resource monitoring. Data is accumulated for the resources that are specified in the MCT (for example, CICS files) and written to SMF.

STATUS(*cvda*)

Specifies whether CICS monitoring is to be active or disabled. CVDA values are as follows:

- OFF** Monitoring is not to occur. No data is accumulated or written out, irrespective of the settings of the monitoring data classes.
- ON** Monitoring is to be active. Data is accumulated for all classes of monitor data, and written out for those classes that are active.

SYNCPOINTST(*cvda*)

Specifies whether CICS is to record performance class data separately for each unit of work (UOW) in tasks that contain multiple UOWs. A UOW in a task ends when a sync point occurs, either explicitly (a SYNCPOINT command) or implicitly (a DL/I TERM call, for example, or task end); a new UOW begins immediately, except at end of task. When rollback occurs on a sync point, the UOW does not end. CVDA values are as follows:

NOSYNCPOINT

Performance data is to be combined over all UOWs in a task.

SYNCPOINT

Performance data is to be recorded separately for each UOW.

TSQUEUELIMIT(*data-value*)

Specifies the maximum number of temporary storage queues for which CICS is to perform transaction resource monitoring, as a halfword binary value. The value specified must be in the range 0 - 64.

Conditions**INVREQ**

RESP2 values:

- 1 STATUS has an invalid CVDA value.
- 2 PERFCLASS has an invalid CVDA value.
- 3 EXCEPTCLASS has an invalid CVDA value.
- 5 CONVERSEST has an invalid CVDA value.
- 6 SYNCPOINTST has an invalid CVDA value.
- 7 The FREQUENCY value is invalid. (The hours exceed 24, minutes or seconds exceed 59, or total value is out of range.)
- 8 The FREQUENCYHRS value is out of range.
- 9 The FREQUENCYMIN value is out of range.
- 10 The FREQUENCYSEC value is out of range.
- 11 COMPRESSST has an invalid CVDA value.
- 12 The FILELIMIT value is out of range.
- 13 The DPLLIMIT value is out of range.
- 14 The TSQUEUELIMIT value is out of range.

NOTAUTH

RESP2 values:

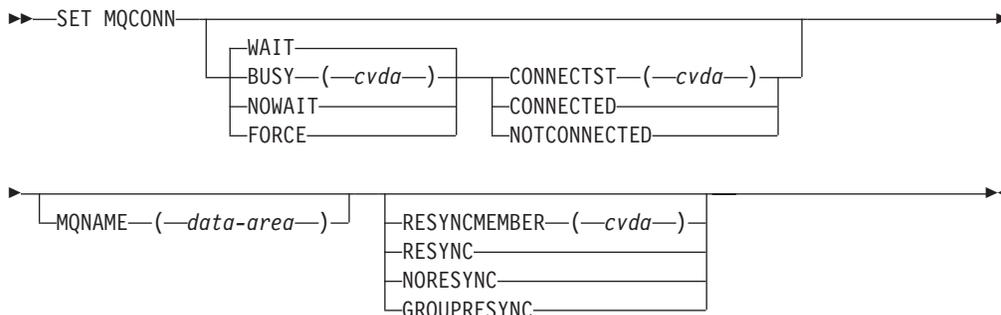
- 15 IDNTYCLASS has an invalid CVDA value.

100 The user associated with the issuing task is not authorized to use this command.

SET MQCONN

Change information about the attributes of the connection between CICS and WebSphere MQ, and start or stop the connection.

SET MQCONN



Conditions: NORMAL, NOTAUTH, NOTFND, INVREQ

This command is threadsafe.

Description

Use the SET MQCONN command to change attributes of the currently installed MQCONN resource definition, which defines the connection to WebSphere MQ, and to start and stop the connection.

Because only one MQCONN resource definition can be installed at a time, the name of the MQCONN resource definition is not required on input.

This command does not set the INITQNAME attribute of the MQCONN resource definition, which specifies the name of the default initiation queue. If you want to change the MQINI resource definition, you must reinstall the MQCONN resource definition with an appropriate MQINI attribute.

Restriction: This command cannot be used in a remote program that is linked by a distributed program link command.

Options

BUSY

This option is valid only with CONNECTST when setting the CICS-WebSphere MQ connection NOTCONNECTED. If you specify CONNECTED, BUSY is ignored. The CVDA values are as follows:

FORCE

Any CICS transactions currently by using WebSphere MQ are abnormally ended, and the connection to WebSphere MQ is stopped. The request is synchronous in nature; that is, control is not returned to the application until the connection is stopped.

NOWAIT

The connection to WebSphere MQ is quiesced. Existing transactions are allowed to finish before the connection is stopped. The request is asynchronous in nature; that is, control is returned to the application before the connection is stopped.

WAIT The connection to WebSphere MQ is quiesced. Existing transactions are allowed to finish before the connection is stopped. The request is synchronous in nature; that is, control is not returned to the application until the connection is stopped. **WAIT** is the default.

CONNECTST

Starts or stops the connection between CICS and WebSphere MQ. The CVDA values are as follows:

CONNECTED

Starts the CICS-WebSphere MQ connection. This action has the same effect as issuing a CKQC START command to start the CICS-WebSphere MQ adapter. If the requested queue manager is active, control returns when CICS and WebSphere MQ are connected. If the requested queue manager is not active, CICS returns a NORMAL response with RESP2=8, indicating that the CICS-WebSphere MQ adapter is in connecting state and reconnects to WebSphere MQ as soon as the requested queue manager becomes active.

NOTCONNECTED

Stops the CICS-WebSphere MQ connection. The value that you specify for the BUSY option determines whether existing transactions are stopped or allowed to complete, and at what stage control is returned to the application. The default is BUSY(WAIT), allowing existing transactions to finish before the connection is stopped, and not returning control to the application until the connection is stopped.

MQNAME

Specifies the 1 - 4 character name of a WebSphere MQ queue manager or queue-sharing group to which CICS is to connect. CICS attempts to connect to the queue manager or to any active member of the queue-sharing group. You can change MQNAME only when CICS is not connected to WebSphere MQ.

When you specify MQNAME, the queue manager name or queue-sharing group that you specified in the MQNAME attribute of the installed MQCONN resource definition is replaced with the name that you specified on this command. If you want to revert to the original queue manager or queue-sharing group, set MQNAME again.

RESYNCMEMBER

This option applies only if you have specified a queue-sharing group for the CICS-WebSphere MQ connection. RESYNCMEMBER specifies the strategy that CICS adopts if outstanding units of work are being held for the last queue manager to which CICS was connected from the queue-sharing group.

Units of work that are shunted indoubt are not included in this process, because CICS itself cannot resolve those units of work at this time. Resynchronization for those UOWs occurs when CICS has resynchronized with its remote coordinator. The CVDA values are as follows:

RESYNC

CICS connects to the same queue manager.

NORESYNC

CICS makes one attempt to connect to the same queue manager. If that

attempt fails, CICS connects to any member of the queue-sharing group and issues a warning message about the outstanding units of work.

GROUPRESYNC

CICS connects to any member of the queue-sharing group. The queue manager is chosen by WebSphere MQ and it asks CICS to resolve indoubt units of work on behalf of all eligible queue managers in the queue-sharing group. This function is called *group unit of recovery*. The GROUPRESYNC option can be used only when you are running a release of WebSphere MQ that supports group unit of recovery for CICS and when the GROUPPUR attribute has been enabled in the WebSphere MQ queue managers.

When an attempt is made to connect CICS to WebSphere MQ by using an **EXEC CICS SET MQCONN CONNECTED** command and RESYNCMEMBER(GROUPRESYNC) is set but WebSphere MQ does not support group unit of recovery, or group unit of recovery is not enabled, then WebSphere MQ rejects the connection attempt. The connection attempt results in the SET command failing with INVREQ and RESP2=9 (connection error).

Do not change the setting for RESYNCMEMBER when units of work are outstanding in WebSphere MQ because this means that units of work cannot be resolved. A unit of work held in CICS is identified with a resource manager qualifier. When RESYNCMEMBER(GROUPRESYNC) is used the qualifier is the name of the queue-sharing group, otherwise the qualifier used is the name of the individual queue manager.

Conditions

NORMAL

RESP2 values:

- 8 Waiting for WebSphere MQ. This situation can occur following a CONNECTST with a CVDA of CONNECT.

NOTAUTH

The user associated with the issuing task is not authorized to use this command, or is not authorized to access this resource.

RESP2 values:

- 100 Command authorization failure.

NOTFND

RESP2 values:

- 1 No MQCONN resource definition is currently installed.

INVREQ

RESP2 values:

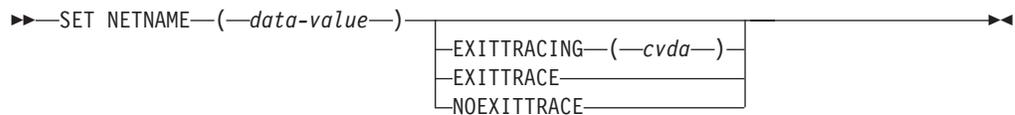
- 2 SET NOTCONNECTED with the FORCE or WAIT option has been specified, but this transaction is itself using the CICS-WebSphere MQ interface.
- 3 MQNAME cannot be set because the connection is active.
- 4 BUSY value is not valid.
- 5 MQNAME contains characters that are not valid.

- 6 CONNECTST value is not valid.
- 7 RESYNCMEMBER value is not valid.
- 9 Connection error.
- 10 MQNAME value is not a valid queue manager or queue-sharing group.

SET NETNAME

Change the tracing of a z/OS Communications Server terminal.

SET NETNAME



Conditions: INVREQ, NOTAUTH

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The SET NETNAME command allows you to control CICS z/OS Communications Server exit tracing for a particular z/OS Communications Server terminal (or session). You can specify any z/OS Communications Server terminal or session, including one not yet installed in CICS, so that you can trace the autoinstall process as well as other operations.

Options

EXITTRACING(*cvda*)

specifies whether this terminal (or session) should be traced when CICS is tracing terminal-specific invocations of its z/OS Communications Server exits. (You can turn exit tracing on and off with a SET TRACEFLAG TCEXITSTATUS command or the CICS-supplied transaction CETR.) CVDA values are:

EXITTRACE

The terminal is to be traced.

NOEXITTRACE

The terminal is not to be traced.

NETNAME(*data-value*)

specifies the 8-character z/OS Communications Server network identifier of the terminal or session for which you are specifying tracing.

Conditions

INVREQ

RESP2 values:

- 27 EXITTRACING has an invalid CVDA value.
- 29 The terminal is not a z/OS Communications Server terminal.

NOTAUTH

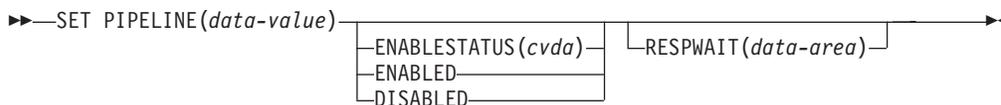
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SET PIPELINE

Change the status of an installed PIPELINE.

SET PIPELINE



Conditions: INVREQ, NOTAUTH, NOTFOUN

This command is threadsafe.

Description

Use the **SET PIPELINE** command to change the status of an installed PIPELINE

Options

PIPELINE(*data-value*)

Specifies the 8-character name of the PIPELINE about which you are inquiring.

ENABLESTATUS(*cvda*)

Specifies the status of the PIPELINE:

ENABLED

Inbound service requests for this PIPELINE are processed normally.

DISABLED

Inbound service requests for this PIPELINE are rejected.

RESPWAIT(*data-area*)

Specifies the number of seconds that an application program should wait for an optional response message from a remote Web service. The value can range from 0 to 9999 seconds. If you do not specify a value, the default timeout value of the transport protocol is used.

- The default timeout value for HTTP is 10 seconds.
- The default timeout value for WebSphere MQ is 60 seconds.

Conditions

INVREQ

RESP2 values:

- 5 An attempt was made to set an invalid value for RESPWAIT.
- 11 An attempt was made to enable or disable an PIPELINE that is in an invalid state

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

NOTFND

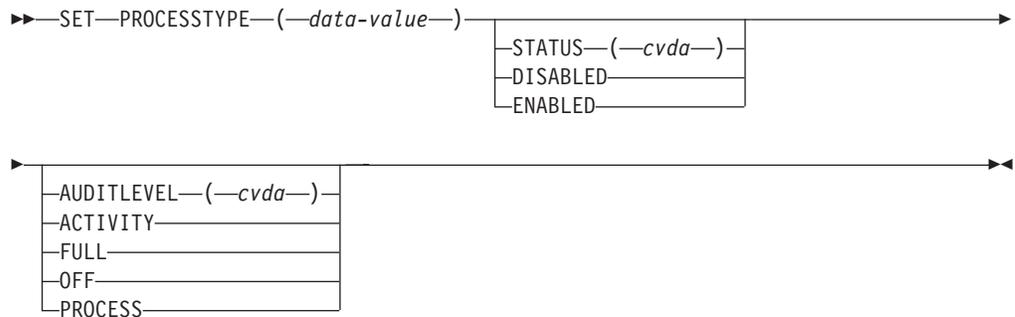
RESP2 values:

3 The PIPELINE was not found.

SET PROCESSTYPE

Change the attributes of a CICS business transaction services (CBTS) process-type.

SET PROCESSTYPE



Conditions: INVREQ, NOTAUTH, PROCESSERR

Description

SET PROCESSTYPE allows you to change the current state of audit logging and the enablement status of PROCESSTYPE definitions installed on this CICS region.

Note: Process-types are defined in the process-type table (PTT). CICS uses the entries in this table to maintain its records of processes (and their constituent activities) on external data sets. If you are using CBTS in a single CICS region, you can use the SET PROCESSTYPE command to modify your process-types. However, if you are using CBTS in a sysplex, it is strongly recommended that you use CPSM to make such changes. This is because it is essential to keep resource definitions in step with each other, across the sysplex.

Options

AUDITLEVEL(cvda)

specifies the level of audit logging to be applied to processes of this type.

Note: If the AUDITLOG attribute of the installed PROCESSTYPE definition is not set to the name of a CICS journal, an error is returned if you try to specify any value other than OFF.

The CVDA values are:

ACTIVITY

Activity-level auditing. Audit records are written from:

1. The process audit points
2. The activity primary audit points.

FULL Full auditing. Audit records are written from:

1. The process audit points
2. The activity primary *and* secondary audit points.

OFF No audit trail records are written.

PROCESS

Process-level auditing. Audit records are written from the process audit points only.

For details of the records that are written from the process, activity primary, and activity secondary audit points, see the *CICS Business Transaction Services*.

PROCESSTYPE(value)

specifies the 8-character name of a process-type defined in the process-type table (PTT), whose attributes are to be changed.

STATUS(cvda)

specifies whether new processes of this type can be created. The CVDA values are:

DISABLED

The installed definition of the process-type is disabled. New processes of this type cannot be defined.

ENABLED

The installed definition of the process-type is enabled. New processes of this type can be defined.

Conditions

INVREQ

RESP2 values:

- 2 The process-type is not disabled, and therefore cannot be enabled.
- 3 You have specified an invalid CVDA value on the AUDITLEVEL option.
- 5 You have specified an invalid CVDA value on the STATUS option.
- 6 You have specified a value of FULL, PROCESS, or ACTIVITY on the AUDITLEVEL option, but the AUDITLOG attribute of the PROCESSTYPE definition does not specify an audit log.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

PROCESSERR

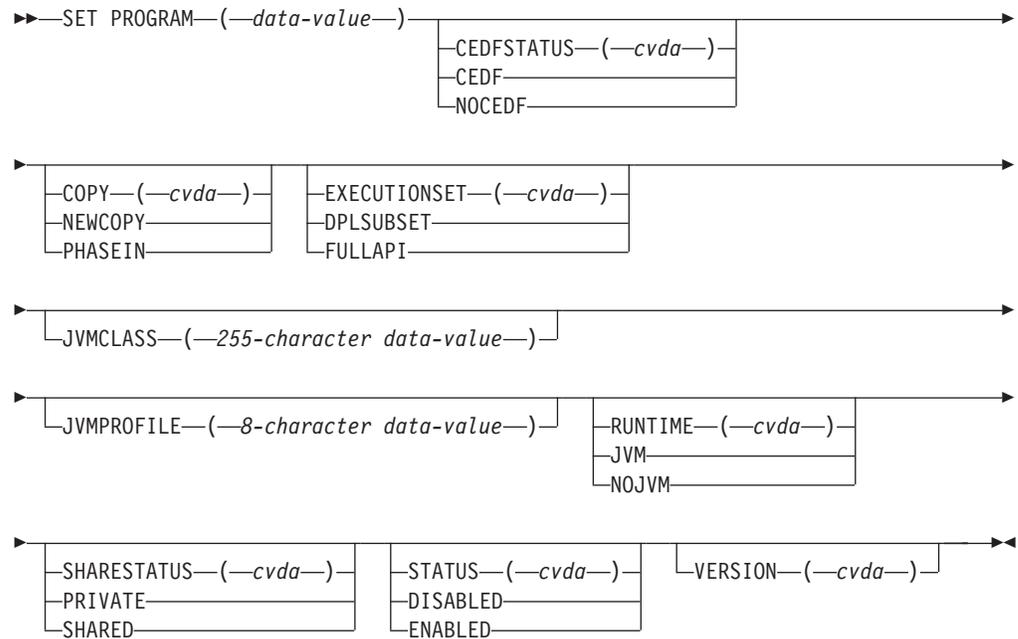
RESP2 values:

- 1 The process-type named in the PROCESSTYPE option is not defined in the process-type table (PTT).

SET PROGRAM

Change a PROGRAM, MAPSET, or PARTITIONSET definition.

SET PROGRAM



Conditions: INVREQ, IOERROR, NOTAUTH, PGMIDERR

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The SET PROGRAM command modifies the definition of a particular program, map set, or partition set installed in your CICS system. All of these resources are load modules and, therefore, CICS uses the same SET command for all three. To avoid confusion, the term *module* means the object of your command, except when the option applies only to executable programs.

Options

CEDFSTATUS(*cvda*) (programs only)

Specifies what action the execution diagnostic facility (EDF) is to take if this program runs under EDF. This option has the following CVDA values:

CEDF EDF diagnostic screens are to be displayed. If the program was translated with the EDF option, all EDF screens are displayed; if it was translated with NOEDF, only the program initiation and termination screens are displayed.

NOCEDF

No EDF screens are displayed.

You cannot specify CEDFSTATUS for a remote program.

COPY(*cvda*)

Specifies that a new copy of the program is to be used the next time the module is requested. **LINK**, **XCTL**, **LOAD**, **ENABLE**, and **BMS** commands can cause a module request.

CICS does not load the module at this time, but it does ensure that a copy is available. If you have specified the **SHARED** option and the module is in the link-pack area, the LPA copy satisfies this requirement. Otherwise, CICS

searches the DFHRPL or dynamic LIBRARY concatenations, and returns an IOERR exception if it cannot locate a copy there.

For Java programs that run in a pooled JVM or JVM server, you cannot use this command to refresh the program. For pooled JVMs, phase out the existing JVMs in the JVM pool using the **PERFORM JVMPOOL TERMINATE** command. When new JVMs are started, they run the new version of the program. For JVM servers, disable and enable the JVMSERVER resource that the Java program uses.

You cannot specify the COPY option for any module currently loaded with the HOLD option, or for any program defined as remote.

This option has the following CVDA values:

NEWCOPY

The module is to be refreshed only if it is not currently in use; otherwise CICS returns an INVREQ exception instead. You can determine whether a module is in use from the RESCOUNT option in an **INQUIRE PROGRAM** command. A value of zero means that the program is not in use.

PHASEIN

The refresh is to occur whether the module is in use. If it is, the copy or copies in use remain until they are no longer in use, but all requests that occur after the refresh use the new copy.

EXECUTIONSET(*cvda*) (programs only)

Specifies whether the program is to be restricted to running the distributed program link (DPL) subset of the CICS API. EXECUTIONSET applies only to executable programs, and governs the API only when a program runs locally. Programs are always restricted to this subset when called remotely; that is, when running at or below the level of a program called by DPL. This option has the following CVDA values:

DPLSUBSET

The program is always to be restricted. You cannot specify this value for CICS-supplied programs (those beginning with 'DFH').

FULLAPI

The program is not to be restricted unless called remotely.

The EXECUTIONSET attribute applies only in the following circumstances:

- Programs that are being linked to and not to those programs which are the first to be given control by a transaction.
- When the REMOTESYSTEM name is the same name as the local CICS region. Its purpose is to test programs in a local CICS environment as if they were running as DPL programs.

JVMCLASS(*255-character data-value*) (Java programs only)

Specifies the name of the main class in the Java program to be given control by the JVM. If you specify JVM in the RUNTIME option, specify a JVMCLASS value. If you specify NOJVM in the RUNTIME option, any value in the JVMCLASS option is ignored when the program runs.

JVMPROFILE(*8-character data-value*) (Java programs only)

Specifies the name of a JVM profile that is to be used for the pooled JVM in which this Java program runs. Any instances of this program that are currently running in a JVM with the old JVM profile are unaffected, and can finish running. You cannot set this option for Java programs that use a JVM server because the JVM profile is set on the JVMSERVER resource.

|
| When you use the name of a JVM profile anywhere in CICS, you must enter it
| using the same combination of uppercase and lowercase characters that are
| present in the z/OS UNIX file name.

PROGRAM(*data-value*)

Specifies the 8-character name of the program, map set, or partition set definition to be changed.

RUNTIME(*cvda*) (**Java programs only**)

Specifies whether the program is to run in a JVM. This option has the following CVDA values:

JVM The program is to run in a JVM. Specify a JVMCLASS value.

NOJVM

The program is not to run in a JVM. Any value in the JVMCLASS option is ignored and the runtime environment of the changed program is unknown until the program is next loaded by CICS, at which point its runtime environment is determined.

SHARESTATUS(*cvda*)

Specifies where CICS obtains the module the next time a new copy is required. A new copy request can result from either an explicit request, for example **SET PROGRAM COPY**, or from a command that requires the module that is issued when CICS does not currently have a copy. This option has the following CVDA values:

PRIVATE

The module is to be loaded from the concatenated libraries named on the DFHRPL or dynamic LIBRARY DD statement.

SHARED

The link-pack area copy is to be used, if one is available. If not, the module is loaded as if SHARESTATUS were PRIVATE.

You cannot specify SHARESTATUS for a remote program. Any value specified for Java programs is ignored.

STATUS(*cvda*)

Specifies whether the module is to be available for use. This option has the following CVDA values:

DISABLED

The module is to be unavailable. CICS programs (beginning with 'DFH') cannot be disabled.

ENABLED

The module is to be available.

For a program defined as remote, this option governs availability only when the program is called through the local CICS system; it does not change availability on the remote system.

The ENABLED and DISABLED option are honored for a program that is called through a CICS program link request, but they have no effect if the program is called by a Java program through a method call.

VERSION(*cvda*)

Returns a CVDA value indicating whether the copy CICS located for a COPY request is different from the current copy. A value is returned only when the COPY option is also specified; in other cases the CVDA value is unchanged. For this purpose, CICS defines "different" to mean a switch from a copy

loaded from the DFHRPL or dynamic LIBRARY concatenations to the link-pack area copy or vice versa, or a copy loaded from a disk location different from that of the current copy. This option has the following CVDA values:

NEWCOPY

The new copy is different.

OLDCOPY

The new copy is not different. This value is always returned for Java programs.

Conditions

INVREQ

RESP2 values:

- 1 DISABLED or DPLSUBSET was specified for a program beginning 'DFH'.
- 2 STATUS has an invalid CVDA value.
- 3 NEWCOPY was specified and RESCOUNT is not equal to zero.
- 4 SHARESTATUS has an invalid CVDA value.
- 5 COPY has an invalid CVDA value.
- 6 COPY was specified for a module currently loaded with the HOLD option.
- 9 CEDFSTATUS has an invalid CVDA value.
- 17 You have specified an option that is invalid for a remote program (CEDFSTATUS, COPY, EXECUTIONSET, or SHARESTATUS).
- 18 You have specified an option that is invalid for a map set (CEDFSTATUS or EXECUTIONSET).
- 19 You have specified an option that is invalid for a partition set (CEDFSTATUS or EXECUTIONSET).
- 20 EXECUTIONSET has an invalid CVDA value.
- 22 RUNTIME has an invalid CVDA value.
- 23 JVM was specified but no JVMCLASS has been supplied.
- 25 JVMCLASS contains embedded blanks or null (x'00') characters.
- 26 JVMPROFILE contains invalid characters or embedded blanks.
- 27 JVMPROFILE cannot be set for a program that uses a JVM server.

IOERR

RESP2 values:

- 8 Either the COPY option or the RUNTIME(NOJVM) option was specified but CICS was not able to locate the module.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

PGMIDERR

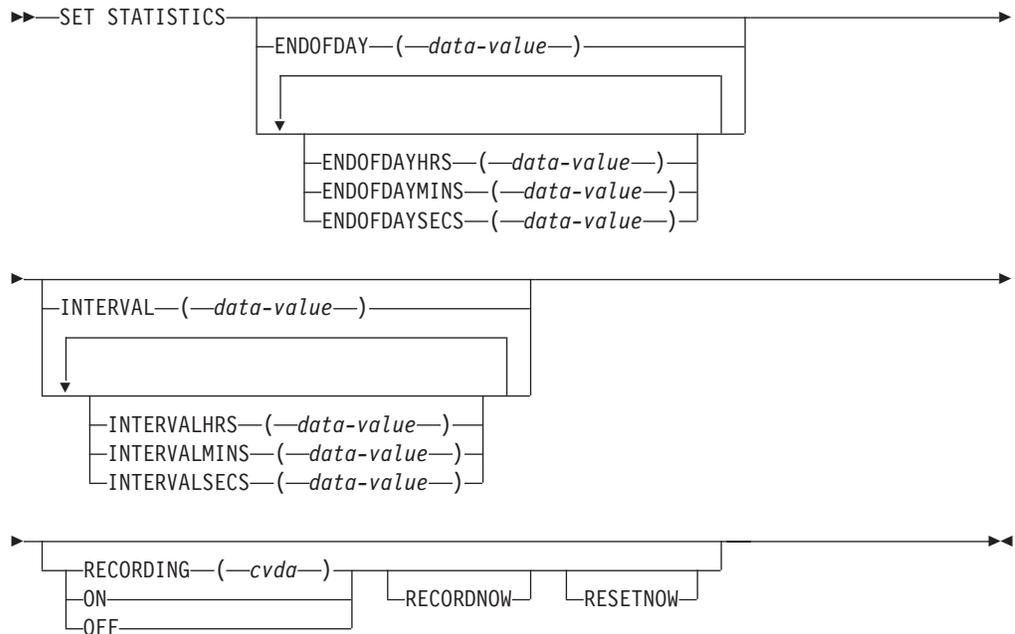
RESP2 values:

- 7 The program, map set, or partition set cannot be found.

SET STATISTICS

Change the recording of CICS statistics.

SET STATISTICS



Conditions: INVREQ, NOTAUTH

For more information about the use of CVDAs, see CICS-value data areas (CVDAs) in CICS System Programming Reference.

Description

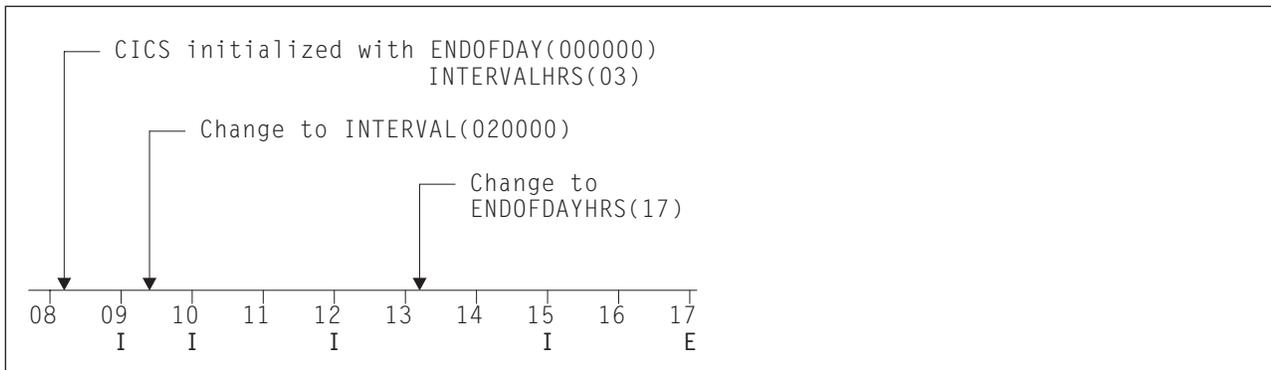
The SET STATISTICS command allows you to change values that control the recording of CICS statistics and to reset the counts.

CICS records system and resource statistics periodically if the RECORDING switch is on, at a frequency governed by the INTERVAL option. These statistics are called **interval statistics**. At end-of-day time (the ENDOFDAY option), CICS records **end-of-day statistics**—which are the statistics for the interval since the last resetting—whether or not the switch is on, ensuring that statistics are written at least once a day. Recording occurs on a system management facility (SMF) data set, and the counts are reset after recording.

When CICS is initialized, the length of the first interval is adjusted so that an integral number of intervals remains until end-of-day time. If you change the recording interval, the same adjustment is made to the current interval. The arrival of end-of-day time, whether changed or not, ends the current recording interval. After the statistics are written out, the next interval is adjusted again if necessary, so that the recording interval divides the time remaining to the next end-of-day evenly.

Note: These adjustments are made whether or not the statistics for the interval get recorded. Consequently, if you want to capture all of the statistics, set RECORDING ON or let your end-of-day recording cover all of them by setting the recording interval to 24 hours.

These rules are illustrated by the following example. I indicates an interval recording and E indicates an end-of-day recording. The system is cold started with STATRCD, the option that sets the initial value for the RECORDING switch, set to ON.



The the *CICS Performance Guide* contains more detail about CICS statistics, including the values to which various types of statistics are re-initialized.

The two time values that you can set with this command can be expressed in several ways:

- A 4-byte packed decimal composite, in the format *Ohhmmss+*, which you specify with the ENDOFDAY or INTERVAL option.
- Separate hours, minutes and seconds, which you specify with the ENDOFDAYHRS, ENDOFDAYMINS, and ENDOFDAYSECS options (instead of ENDOFDAY) and INTERVALHRS, INTERVALMINS, and INTERVALSECS (instead of INTERVAL). You can use these options singly or in any combination.

For example, you could express an INTERVAL of 1 hour and 30 minutes in any of the following ways:

- INTERVAL(13000)
- INTERVALHRS(1), INTERVALMINS(30)
- INTERVALMINS(90)
- INTERVALSECS(5400)

Options

ENDOFDAY (*data-value*)

specifies the end-of-day time, as a 4-byte packed decimal field in the format *Ohhmmss+*.

End-of-day time is expressed in local time and must be in the range 00:00:00-23:59:59. When you use the ENDOFDAY option, or more than one of the separate end-of-day options, neither the minutes nor the seconds portions can exceed 59. If you use ENDOFDAYMINS alone the limit is 1439 and for ENDOFDAYSECS used alone it is 86399.

ENDOFDAYHRS (*data-value*)

specifies the hours component of the end-of-day time, in fullword binary form. (See the ENDOFDAY option.)

ENDOFDAYMINS (*data-value*)

specifies the minutes component of the end-of-day time, in fullword binary form. (See the ENDOFDAY option.)

ENDOFDAYSECS(*data-value*)

specifies the seconds component of the end-of-day time, in fullword binary form. (See the ENDOFDAY option.)

INTERVAL(*data-value*)

specifies the recording interval for system statistics, as a 4-byte packed decimal field in the format *Ohhmmss+*. The interval must be at least a minute and no more than 24 hours. When you use the INTERVAL option, or more than one of the separate interval options, neither the minutes nor the seconds portions of the time must exceed 59. If you use INTERVALMINS alone the range is 1-1440 and for INTERVALSECS used alone it is 60-86400.

INTERVALHRS(*data-value*)

specifies the hours component of the recording interval, in fullword binary form. (See the INTERVAL option.)

INTERVALMINS(*data-value*)

specifies the minutes component of the recording interval, in fullword binary form. (See the INTERVAL option.)

INTERVALSECS(*data-value*)

specifies the seconds component of the recording interval, in fullword binary form. (See the INTERVAL option.)

RECORDING(*cvda*)

specifies whether to record the interval statistics. End-of-day statistics, requested statistics and unsolicited statistics are always recorded, irrespective of the setting of the RECORDING option. (Statistics are always accumulated, and end-of-day, unsolicited, and requested statistics always recorded, regardless of the setting of the RECORDING option. Unsolicited statistics are resource statistics, recorded when the resource is discarded. Requested statistics are those called for by a PERFORM STATISTICS RECORD command, or by a CEMT PERFORM STATISTICS transaction.)

CVDA values are:

OFF switches off the recording of interval statistics.

ON switches on the recording of interval statistics.

RECORDNOW

specifies that the current statistics are to be written out immediately. The effect is the same as a PERFORM STATISTICS RECORD ALL command and, as in the case of that command, the counts are not reset unless you specify RESETNOW as well. RECORDNOW can be specified only when the RECORDING status is changed from ON to OFF or from OFF to ON.

RESETNOW

specifies that the statistics counters are to be reset to their initial values. The initial value for a given counter depends on the type of statistic being collected; see the *CICS Performance Guide* for specific information. The reset can be requested only when the RECORDING status is changed from ON to OFF or from OFF to ON.

Conditions**INVREQ**

RESP2 values:

- 1 The INTERVAL value is out of range.
- 2 The ENDOFDAY value is out of range.

- 3 RECORDING has an invalid CVDA value.
- 4 The INTERVALHRS value is out of range.
- 5 The INTERVALMINS value is out of range.
- 6 The INTERVALSECS value is out of range.
- 7 More than one of the interval values has been used and the combination either exceeds 24 hours or is less than 1 minute.
- 8 The ENDOFDAYHRS value is out of range.
- 9 The ENDOFDAYMINS value is out of range.
- 10 The ENDOFDAYSECS value is out of range.
- 11 RESETNOW or RECORDNOW has been specified, but the RECORDING value has not been changed.

NOTAUTH

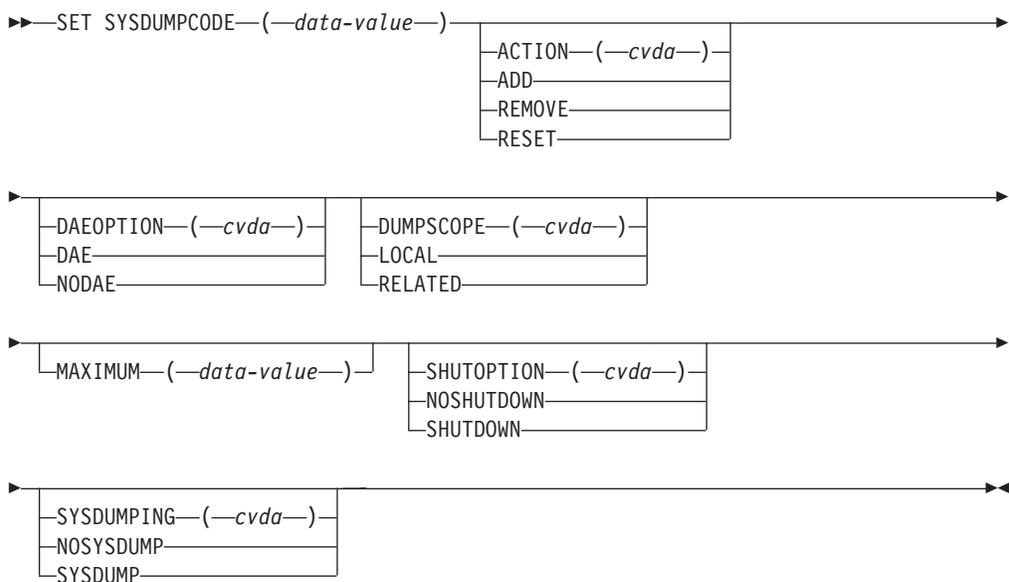
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SET SYSDUMPCODE

Change an entry in the system dump table.

SET SYSDUMPCODE



Conditions: DUPREC, INVREQ, IOERR, NOSPACE, NOTAUTH, NOTFND

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The **SET SYSDUMPCODE** command allows you to change the system dump table entry for a particular dump code, to add a new dump code to the table, or to delete one.

The table entry tells CICS the actions to take when a system dump request with this code occurs. Possible actions include taking a system dump (an MVS SDUMP), initiating requests for SDUMPs of related CICS regions, and shutting down CICS. The table entry also indicates how many times this set of actions is to be taken (the MAXIMUM value); after the maximum is reached, requests are counted but otherwise ignored.

Table updates are recorded in the CICS global catalog and preserved over executions of CICS until an initial or cold start occurs, except in the case of temporary table entries. CICS creates a temporary entry when it receives a dump request with a code for which there is no table entry; these entries, and any changes to them, last only for the current execution of CICS. If you want to preserve changes to a temporary entry over restarts, you must remove the dump code from the table and then add it back.

For information about system dumps, see [The dump code options you can specify](#).

Options

ACTION(*cvda*)

Specifies the action to be taken for the dump code. CVDA values are:

ADD An entry for this code is to be added to the table.

REMOVE

The entry for this code is to be removed from the table. No other option can be specified on a **SET SYSDUMPCODE REMOVE** command.

RESET

The current number of dump requests for this code is to be set to zero. See the CURRENT option of the **INQUIRE SYSDUMPCODE** command.)

DAEOPTION

Specifies whether a dump produced for this dump code is eligible for suppression by the MVS Dump Analysis and Elimination (DAE) component. CVDA values are:

DAE The dump is eligible for DAE suppression.

NODAE

The dump is not eligible for DAE suppression. If CICS determines that a dump should be written, MVS does not suppress it. However, the SUPPRESS and SUPPRESSALL options in the ADYSETxx parmlib member are controlled by the VRADAE and VRANODAE keys in the SDWA. They might lead to dump suppression even though NODAE is set here. For information about these options, see *z/OS MVS Diagnosis: Tools and Service Aids*.

When **SET SYSDUMPCODE ADD** is specified, if you do not also specify DAEOPTION, it defaults to NODAE, regardless of the setting of the **DAE** system initialization parameter.

DUMPSCOPE(*cvda*)

Specifies whether a request for a dump with this dump code causes CICS to initiate requests for SDUMPs (system dumps) of “related” CICS regions.

A related CICS region is one in the same sysplex, connected by MRO/XCF and doing work on behalf of your CICS region - specifically, a region that has one or more tasks doing work under the same APPC token as a task in your region.

This propagation of SDUMP requests occurs only when the table entry for this code also specifies a SYSDUMPING value of SYSDUMP, and only in a sysplex environment.

If you specify RELATED in other systems, this causes an exception condition.

CVDA values are:

LOCAL

SDUMP requests are not to be sent.

RELATED

SDUMP requests are to be sent.

Note: A setting of DUMPSCOPE(RELATED) results in a single dump being taken for each affected MVS image. This dump contains the output from all the affected CICS regions in the image. For more information, see Automatic dump data capture from related CICS regions.

LOCAL is the default for entries you add, if you do not specify a DUMPSCOPE value.

MAXIMUM(*data-value*)

Specifies, as a fullword binary value, the maximum number of dumps with this code that CICS should request, in the range 0-999. After the maximum is reached, CICS counts but otherwise ignores dump requests with this code. A value of 999 means there is no limit, and is the default for new entries if you do not specify a MAXIMUM value.

SHUTOPTION(*cvda*)

Specifies whether the system is to be shut down after a request for a dump with this dump code. CVDA values are:

NOSHUTDOWN

The system is not to be shut down. This value is assumed if you omit it from a **SET SYSDUMPCODE ADD** command.

SHUTDOWN

The system is to be shut down.

SYSDUMPCODE(*data-value*)

Specifies the 8-character system dump code for which the system dump table entry is to be modified. A valid system dump code contains no leading or imbedded blanks.

SYSDUMPING(*cvda*)

Specifies whether a system dump request with this code should produce a dump. CVDA values are:

NOSYSDUMP

A dump is not to be taken.

SYSDUMP

A dump is to be taken.

Even when you specify SYSDUMP, CICS takes a dump only if the number of requests for this code is less than the MAXIMUM and system dumps are not suppressed globally (see the DUMPING option of the **INQUIRE SYSTEM** command). MVS may also be allowed to suppress the dump if appropriate, depending on the DAEOPTION value.

If the SYSDUMPING option is omitted from a **SET SYSDUMPCODE ADD** command, SYSDUMP is assumed.

Conditions

DUPREC

RESP2 values:

- 10 ADD is specified for a dump code already in the system dump table.

INVREQ

RESP2 values:

- 2 ACTION has an invalid CVDA value.
- 4 SYSDUMPING has an invalid CVDA value.
- 5 The MAXIMUM value is out of range.
- 6 SHUTOPTION has an invalid CVDA value.
- 7 REMOVE is specified with other options.
- 9 The dump code is invalid.
- 13 DUMPSCOPE has an invalid CVDA value.
- 15 DAEPTION has an invalid CVDA value.

IOERR

RESP2 values:

- 11 An error occurred updating the CICS catalog. The entry is changed for the current run, but is not recorded for restarts.

NOSPACE

RESP2 values:

- 12 The CICS catalog is full. The entry is changed for the current run, but is not recorded for restarts.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

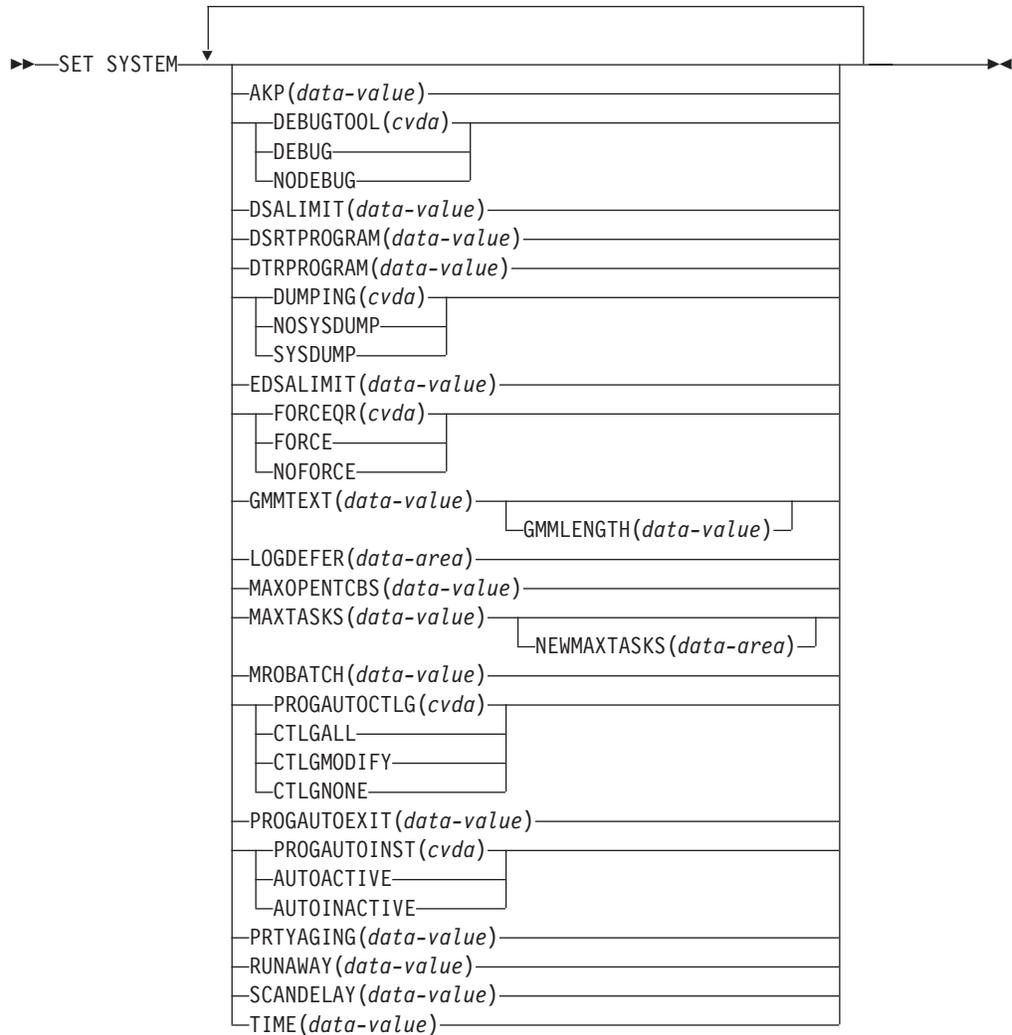
RESP2 values:

- 1 The dump code cannot be found.

SET SYSTEM

Change CICS system option values.

SET SYSTEM



Conditions: INVREQ, LENGERR, NOSTG, NOTAUTH, NOTSUPERUSER

Note:

1. For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.
2. The CSCS, ECSCS, USCS, EUSCS, and ERSCS options, each of which returned the size of the storage “cushion” for a particular dynamic storage area, are obsolete. To maintain object compatibility, they are accepted at run time but ignored. The translator also accepts them, but issues a warning message.

Description

Use the SET SYSTEM command to change the values of some options that control the execution of your CICS system.

These values are set initially by system initialization parameters. System initialization parameters that correspond to those in this command have the same or similar names, except where noted. Table 4 on page 441 lists the exact correspondence.

Options

AKP(*data-value*)

Specifies, as a fullword binary value, the activity keypoint trigger value, which is the number of write requests to the CICS system log stream output buffer between the taking of keypoints. The number must be either zero, which turns off keypointing, or in the range 200–65535. If CICS was initialized without keypointing (that is, with the AKPFREQ system initialization parameter set to zero), the initial value can be overridden and a trigger value can be set.

DEBUGTOOL(*cvda*)

Specifies a CVDA value indicating whether you want to use debugging profiles to select the programs that will run under the control of a debugging tool. The following debugging tools use debugging profiles:

- Debug Tool, for compiled language application programs (programs written in COBOL, PL/I, C, C++ and Assembler)
- Remote debugging tools (for compiled language application programs and Java programs)

Other debugging mechanisms, such as the CICS Execution Diagnostic Facility (CEDF) do not use debugging profiles.

The CVDA values are as follows:

DEBUG

Specifies that you want to use CICS debugging profiles to select the programs that will run under the control of a debugging tool.

NODEBUG

Specifies that you do not want to use CICS debugging profiles to select the programs that will run under the control of a debugger tool.

For more information about using debugging profiles, see the *CICS Customization Guide*.

DSALIMIT(*data-value*)

Specifies, as a fullword binary value, the maximum amount of storage, in bytes, within which CICS can allocate storage for the four individual dynamic storage areas (DSAs) that reside below 16 MB (below the line). If DSALIMIT specifies a value lower than the current limit, CICS might not implement the new limit immediately, but attempts to do so over time as storage is freed. The range for DSALIMIT is 2 - 16 MB. For more information, see DSALIM system initialization parameter in the System Definition Guide.

DSRTPROGRAM(*data-value*)

Specifies the 8-character name of the distributed routing program.

DTRPROGRAM(*data-value*)

Specifies the 8-character name of the dynamic routing program.

DUMPING(*cvda*)

Specifies a CVDA value indicating whether the taking of CICS system dumps is to be suppressed. CVDA values are as follows:

NOSYSDUMP

System dumps are to be suppressed.

SYSDUMP

System dumps are not to be suppressed.

EDSALIMIT(*data-value*)

Specifies, as a fullword binary value, the maximum amount of storage, in

bytes, within which CICS can allocate storage for the individual dynamic storage areas that reside above 16 MB but below 2 GB (above the line). If EDSALIMIT specifies a value lower than the current limit, CICS might not implement the new limit immediately, but attempts to do so over time as storage is freed. For more information, see EDSALIM system initialization parameter in the System Definition Guide.

FORCEQR(*cvda*)

Specifies whether you want CICS to force all user application programs specified as CONCURRENCY(THREADSAFE) to run under the CICS QR TCB, as if they were specified as CONCURRENCY(QUASIRENT) programs.

This allows you, in a test environment, to run incompletely tested threadsafe application programs that have proved to be non-threadsafe.

FORCEQR applies to all programs defined as threadsafe that are not invoked as task-related user exits, global user exits, or user-replaceable modules.

CVDA values are as follows:

FORCE

Force all CICSAPI user application programs to run under the QR TCB, even if they are defined with the CONCURRENCY(THREADSAFE) attribute. Force does not apply to certain programs, for example OPENAPI programs, or C or C++ programs compiled with XPLINK. For details, see FORCEQR system initialization parameter in the System Definition Guide.

NOFORCE

CICS is to honor the CONCURRENCY(THREADSAFE) attribute defined on program resource definitions, and invoke them under either the QR TCB or an open TCB.

The FORCEQR(FORCE|NOFORCE) option allows you to change dynamically the option specified by the **FORCEQR** system initialization parameter.

Specifying FORCEQR(FORCE) is not applied to currently invoked programs, and applies only to programs invoked for the first time after the change to the FORCEQR status.

GMMLENGTH(*data-value*)

Specifies, as a halfword binary value, the length of the “good morning” message text. The range for this value is 1–246.

GMMTEXT(*data-value*)

Specifies the “good morning” message text, which can be up to 246 characters long.

LOGDEFER(*data-area*)

Specifies, as a halfword binary value, the log deferral interval, which is the period of time used by CICS Log Manager when determining how long to delay a forced journal write request before invoking the MVS system logger. The value can be in the range 0–65535. For information about the **LOGDEFER** parameter and associated **LGDFINT** system initialization parameter, see LGDFINT system initialization parameter in the System Definition Guide.

MAXOPENTCBS(*data-value*)

Specifies, as a fullword binary value, the maximum number of open TCBS that can exist concurrently in the CICS region. The value specified can be in the range 1 to 999.

If you reduce MAXOPENTCBS from its previously defined value, and the new value is less than the number of open TCBs currently allocated, CICS detaches TCBs to achieve the new limit only when they are freed by user tasks. Transactions are not abended to allow TCBs to be detached to achieve the new limit.

If there are tasks queued waiting for an open TCB, and you increase MAXOPENTCBS from its previously defined value, they are resumed when CICS attaches new TCBs up to the new limit.

MAXTASKS (*data-value*)

Specifies, as a fullword binary value, the maximum number of tasks that can be eligible for dispatch at any one time in this CICS system. Both active and suspended tasks count toward this limit, but tasks that have not reached the point of initial dispatch do not. System tasks such as terminal and journal control tasks do not count either. The value can be in the range 1–999.

MROBATCH (*data-value*)

Specifies, as a fullword binary value, the number of events that must occur, from a list of MRO and DASD I/O events on which CICS is waiting, before CICS is posted explicitly to process them. The value must be in the range 1–255.

NEWMAXTASKS (*data-area*)

Returns the new value of MAXTASKS, in fullword binary form.

When you set MAXTASKS in a SET SYSTEM command, CICS adjusts the value you specify downward if there is not enough storage for the value you request; NEWMAXTASKS tells you what the value is after any such adjustment. CICS also raises the NOSTG condition when it reduces the value, although it continues processing your command.

PROGAUTOCTLG (*cvda*)

Specifies whether and when autoinstalled program definitions are to be cataloged. Cataloged definitions are restored on a warm or emergency restart. Those not cataloged are discarded at shutdown and must be installed again if used in a subsequent execution of CICS.

Decisions to catalog are made both at initial installation and whenever an autoinstalled definition is modified, and are based on the PROGAUTOCTLG value at the time. CVDA values are as follows:

CTLGALL

Definitions are to be cataloged when installed and when modified.

CTLGMODIFY

Definitions are to be cataloged only when modified.

CTLGNONE

Definitions are not to be cataloged.

PROGAUTOEXIT (*data-value*)

Specifies the 8-character name of the user-provided program to be called by the CICS program autoinstall code to provide a model definition.

Note: This program (and any programs it invokes) must be installed before they can be used in the program autoinstall process, either by explicit PROGRAM definitions or by autoinstall when some other autoinstall program is in force. Otherwise, the program autoinstall process fails when next used, and CICS makes it inactive.

PROGAUTOINST(*cvda*)

Specifies whether autoinstall for programs is to be active or inactive. When a task requests a program, map set, or partition set that is not defined, CICS attempts to create a definition for it automatically if autoinstall for programs is active. If not, CICS raises the PGMIDERR exceptional condition. CVDA values are as follows:

AUTOACTIVE

Autoinstall for programs is to be active.

AUTOINACTIVE

Autoinstall for programs is to be inactive.

PRTYAGING(*data-value*)

Specifies, as a fullword binary value, the rate at which CICS is to increase the priority of a task waiting for dispatch. CICS increases the task priority by 1 after each PRTYAGING milliseconds of wait time without a dispatch. The value must be in the range 0–65535.

RUNAWAY(*data-value*)

Specifies, as a fullword binary value, the default for runaway task time. This value is used for any task executing a transaction with a profile that does not specify runaway task time (see the RUNAWAY option of the **INQUIRE TRANSACTION** command on page INQUIRE TRANSACTION options - RTIMEOUT).

The value must be either zero, which means that runaway task detection is not required for tasks using the default value, or in the range 500–2700000. The value you supply is rounded down to the nearest multiple of 500.

SCANDELAY(*data-value*)

Specifies, as a fullword binary value, the maximum number of milliseconds between a user task making a terminal I/O request and CICS dispatching the terminal control task to process it. This value is sometimes called the “terminal scan delay”, and is set initially by the system initialization option ICVTSD. The value must be in the range 0–5000.

TIME(*data-value*)

Specifies, as a fullword binary value, the maximum interval in milliseconds for which CICS gives control to the operating system if no tasks are ready for dispatch. This value is set initially by the ICV system initialization option and is sometimes called the “region exit time interval”. The TIME value must be in the range 100–3600000 and must not be less than the SCANDELAY value. You can determine the current SCANDELAY value, if you are not setting it at the same time, with the **INQUIRE SYSTEM SCANDELAY** command.

Conditions**INVREQ**

RESP2 values:

- 1 The MAXTASKS value is out of range.
- 3 The AKP value is out of range.
- 5 TIME is not in the range 100–3600000.
- 6 The RUNAWAY value is out of range.
- 7 MROBATCH is not in the range 1–255.
- 9 DUMPING has an invalid CVDA value.

- 12 AKP was specified, but CICS was initialized without keypointing.
- 13 TIME is less than SCANDELAY.
- 14 PRYAGING is not in the range 0–65535.
- 15 SCANDELAY is not in the range 0–5000.
- 20 DSALIMIT is not in the range 2 to 16 MB.
- 21 EDSALIMIT is not in the range 48MB to 2047MB.
- 22 There is insufficient MVS storage to allocate DSALIMIT.
- 23 There is insufficient MVS storage to allocate EDSALIMIT.
- 29 The LOGDEFER value is out of range.
- 30 MAXSOCKETS is not in the range 1 through 65535.

LENGERR

RESP2 values:

- 20 The GMMLENGTH value is out of range.

NOSTG

RESP2 values:

- 16 CICS reduced the value you requested for MAXTASKS because of storage constraints; see the NEWMAXTASKS option.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTSUPERUSER

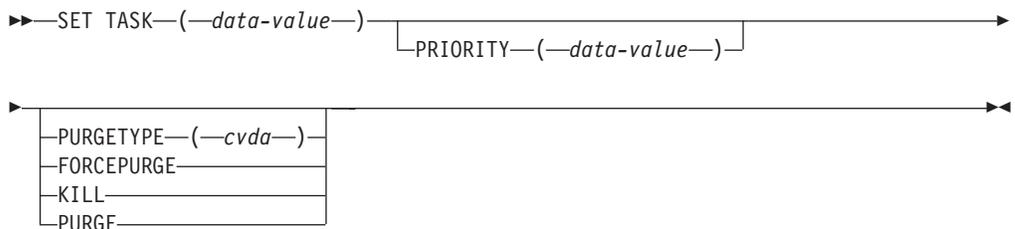
RESP2 value:

- 15 CICS was unable to set MAXSOCKETS to the value you requested, because the userid under which the CICS job is running does not have superuser authority. CICS has set the limit to the value of the MAXFILEPROC parameter specified in SYS1.PARMLIB member BPXPRMxx.

SET TASK

Purge a task or change its priority.

SET TASK



Conditions: INVREQ, NORMAL, NOTAUTH, TASKIDERR

Description

The SET TASK command allows you to purge a task (terminate it abnormally) or to change its priority. Not all tasks can be changed with this command, however; in particular, CICS-created tasks that are essential to system operation are ineligible.

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Options

PRIORITY(*data-value*)

specifies, as a fullword binary value, the priority you want for the task. The value must be in the range 0–255.

PURGETYPE(*cvda*)

specifies that CICS is to purge the task, and indicates conditions for doing so.

Purging a task at the wrong time can result in a loss of data integrity or, in some circumstances, can cause CICS to abend. CICS always defers purging until the task reaches a state where the system itself does not appear to be at risk, but you can specify whether CICS also should wait until data integrity can be ensured.

If CICS accepts a purge request, it returns a NORMAL response to SET TASK. You can tell whether execution has been deferred by inspecting the RESP2 value. If RESP2 is zero, the purge has been completed; if RESP2 is 13, it has been deferred. CVDA values are:

FORCEPURGE

The task is to be terminated as soon as it is consistent with system integrity and without regard to data integrity.

Note: CICS cannot always determine whether a forced purge is safe; it is possible to abend the system when you specify FORCEPURGE.

Kill

The task is to be terminated. System and data integrity is not guaranteed. The KILL option extends the PURGE and FORCEPURGE options. It should be used only after an attempt has been made to PURGE or FORCEPURGE a task. The KILL option does not guarantee integrity of any kind, but in some situations it allows the user to free up a stalled region enabling the region to continue processing. In some cases, for example, if a task is killed during backout processing, CICS terminates abnormally.

PURGE

The task is to be terminated as soon as both system and data integrity can be maintained.

Note: You cannot purge a task with this CVDA value if the definition of the TRANSACTION it is executing specifies SPURGE=NO.

TASK(*data-value*)

specifies the 4-byte packed-decimal sequence number of the task you are changing.

Description

The SET TCLASS command allows you to set the maximum number of tasks in a particular transaction class that are allowed to run concurrently.

This command is limited to the numbered classes of earlier releases of CICS and is retained for compatibility with those releases. The SET TRANCLASS command, described on page “SET TRANCLASS” on page 751, provides the same function and can be used for either the old numbered or new named classes.

Options

MAXIMUM(*data-value*)

specifies, as a fullword binary value, the largest number of tasks in the transaction class that are allowed to run concurrently. The value can be in the range 0 through 999. (This value corresponds to the MAXACTIVE value in a SET TRANCLASS command. See the description of this option on page “SET TRANCLASS” on page 751 for a description of what happens when you change the MAXACTIVE limit.)

TCLASS(*data-value*)

specifies, as a fullword binary value, the number of the task class that you are changing. It must be in the range 0–10.

Conditions

INVREQ

RESP2 values:

2 The MAXIMUM value is not in the range 0-999.

NOTAUTH

RESP2 values:

100 The user associated with the issuing task is not authorized to use this command.

TCIDERR

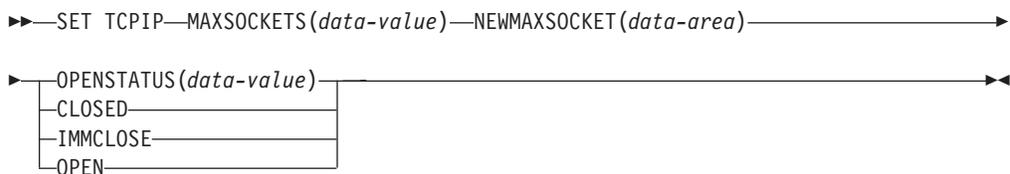
RESP2 values:

1 The transaction class cannot be found.

SET TCPIP

Modify CICS internal TCPIP support.

SET TCPIP



Conditions: INVREQ, IOERR, NOTAUTH, NOTSUPERUSER

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

SET TCPIP allows you to open or close CICS internal sockets support.

Options

MAXSOCKETS (*data-value*)

specifies, as a fullword binary field, the maximum number of IP sockets that can be managed by the CICS sockets domain.

If the userid under which the CICS job is running has superuser authority, up to 65535 sockets can be managed by the sockets domain.

If the userid under which the CICS job is running does not have superuser authority, the maximum number of sockets that can be managed by the sockets domain is limited to the number specified in the MAXFILEPROC parameter in SYS1.PARMLIB member BPXPRMxx. If you specify a greater value, CICS sets the limit to MAXFILEPROC.

Note that sockets created by Java programs running on threads that are not managed by CICS do not count towards the MAXSOCKETS limit.

If you reduce the limit to less than the number of sockets currently active, CICS prevents new sockets from being created until the number of active sockets falls below the limit.

NEWMAXSOCKET (*data-area*)

returns, in a fullword binary field, the new value of MAXSOCKETS.

If the userid under which the CICS job is running does not have superuser authority, CICS may set the MAXSOCKETS limit to a smaller value than requested. NEWMAXSOCKET tells you the limit that CICS has set.

OPENSTATUS (*cvda*)

specifies whether TCPIP is to be enabled (that is, able to process new incoming work, and complete ongoing work), and if TCPIP support is to be disabled, how the disable should be done. CVDA values are:

OPEN

CICS internal TCPIP support is to be opened.

CLOSED

CICS internal sockets support is to be closed. If it is currently open, CICS is to quiesce all internal sockets activity and then close any sockets on which CICS is listening for incoming CICS Web Interface work. Tasks using CICS internal sockets are allowed to complete.

IMMCLOSE

CICS internal sockets is to be closed. If it is currently enabled, CICS is to terminate abnormally any tasks using it and then close the socket on which CICS is listening for incoming work.

Conditions

INVREQ

RESP2 values:

- 4 TCPIP not available (TCPIP=NO)
- 6 TCPIP already open (for SET OPEN)
- 11 STATUS has an invalid CVDA value.
- 12 The OPEN request did not complete because another task subsequently requested a CLOSE of CICS internal sockets support.
- 16 MAXSOCKETS is not in the range 1 to 65535

NORMAL

RESP2 value:

- 14 TCPIP has been opened, but some TCPIP SERVICES have not been opened because the MAXSOCKETS limit has been reached.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTSUPERUSER

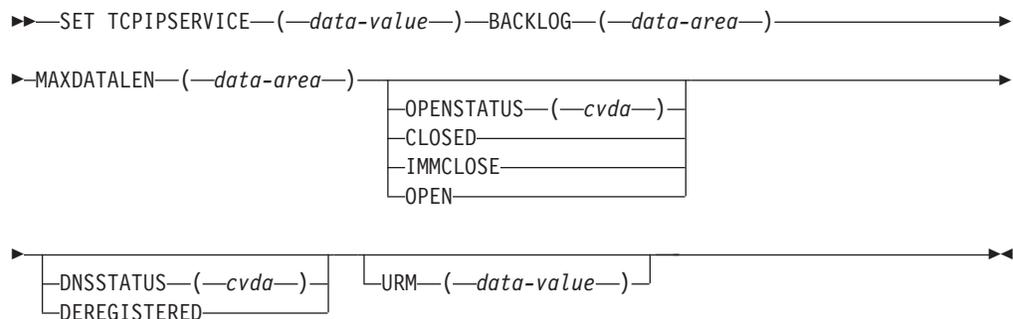
RESP2 values:

- 15 CICS was unable to set the MAXSOCKETS limit to the value requested, because the userid which the CICS job is running under does not have superuser authority. CICS has set the limit to the value of the MAXFILEPROC parameter specified in the SYS1.PARMLIB member BPXPRMxx.

SET TCPIP SERVICE

Modify the status of a service using CICS internal TCP/IP support.

SET TCPIP SERVICE



Conditions: INVREQ, NOTAUTH, NOTFND

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

SET TCPIP SERVICE tells CICS to start or stop listening for incoming requests on the port associated with a service using CICS internal TCP/IP support, and changes the attributes of the service.

Note: This command has no effect on the sockets support provided by the TCP/IP for CICS Sockets Feature.

Options

BACKLOG(*data-area*)

Changes the maximum number of requests that can be queued in TCP/IP waiting to be processed by the service. The service must be CLOSED before you can change this value. If you set BACKLOG to zero, CICS does not receive any connections.

DNSSTATUS

Changes the Domain Name System (DNS) /Workload Manager (WLM) registration status of this service. This can be done independently of changing the open or closed status of the service.

To account for timing delays in the deregister request reaching the WLM and the DNS updating its tables, it is advisable to deregister a service before setting it closed. This ensures that client applications do not encounter "Connection Refused" situations during the time between the deregister call being issued and the DNS server updating its tables. The valid CVDA values are:

DEREGISTERED

Causes CICS to deregister the group name specified by the DNSGROUP attribute of this TCPIPSERVICE. The WLM macro IWMSRDRS is called and CICS is no longer a part of the DNS connection optimization. Any other TCPIP SERVICES that are in the same group (that is, share the same DNSGROUP attribute) are also deregistered.

OPENSTATUS(*cvda*)

Changes the status of the service. CVDA values are:

OPEN

CICS internal TCP/IP support is to start listening for incoming requests on the specified port.

CLOSED

CICS internal sockets support is to stop listening for incoming work on the specified port.

IMMCLOSE

CICS internal sockets support is to stop listening for incoming work on the specified port. If it is currently active, CICS is to terminate abnormally any tasks using it.

MAXDATALEN(*data-area*)

Changes the maximum length of data, in kilobytes, that may be received by CICS as an HTTP server. The default value is 32. The minimum is 3 and the maximum is 524288.

URM(*data-value*)

Specifies the 8-character name of the program to be used as the Service User-replaceable module. You can specify either an installation-specific program or the CICS-supplied default for the service. Some services might not allow this name to be changed.

Conditions

INVREQ

RESP2 values:

- 4 TCP/IP is not available (TCPIP=NO)
- 5 TCP/IP status is closed
- 7 Port is in use
- 8 CICS is not authorized to use this port
- 9 TCPIP SERVICE not closed
- 10 Unknown IP address
- 11 Invalid value specified in an operand of the SET command.
- 12 The requested action cannot be performed because the openstatus of the service does not allow it.
- 13 TCP/IP is inactive.

- 14 The TCPIP SERVICE has not been opened because the MAXSOCKETS limit has been reached.
- 17 The TCPIP SERVICE cannot be opened because PROTOCOL(IIOP) was specified and the **IIOPLISTENER** system initialization parameter is set to NO.
- 19 The TCPIP SERVICE cannot be opened because the IP address or host is not known.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

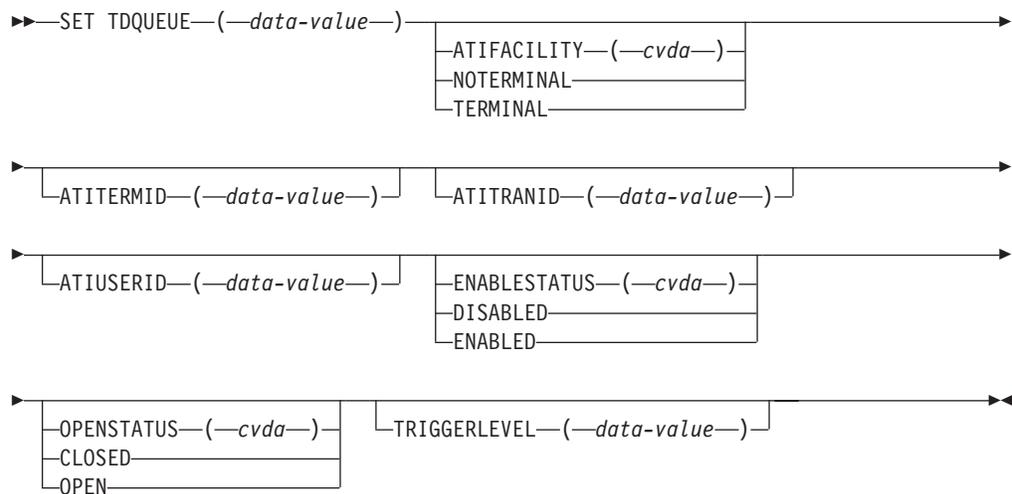
RESP2 values:

- 3 The named TCPIP SERVICE is not found.

SET TDQUEUE

Change the attributes of a transient data queue.

SET TDQUEUE



Conditions: INVREQ, IOERR, NOTAUTH, QIDERR, USERIDERR

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.

Description

The SET TDQUEUE command allows you to change some attributes of a transient data queue.

Transient data queues, also called destinations, are defined in TDQUEUE resource definitions. There are two basic types: **intrapartition** and **extrapartition**. Intrapartition queues are managed and stored entirely by CICS, and are eligible for automatic task initiation (ATI), the facility that CICS provides for scheduling tasks automatically. For a transient data queue, ATI is governed by the value specified on the TRIGGERLEVEL option. If the value is nonzero, CICS automatically creates

a task to process the queue when the number of items on the queue reaches this trigger level. A value of zero exempts the queue from ATI.

An extrapartition queue is an MVS sequential data set (or a spool file). Extrapartition queues are not subject to ATI.

There are two other types of queue: **indirect** and **remote**, both of which point to one of the basic types. You cannot modify the definition of either with a SET TDQUEUE command, however. (See the INQUIRE TDQUEUE command for more information about these queues.)

You cannot alter the following parameters unless the queue is fully disabled:

- ATIFACILITY
- ATITERMID
- ATITRANID.
- ATIUSERID

To disable a transient data destination, the queue must not currently be in use. If it is in use, the queue enters a “disable pending” state. The last unit of work (UOW) to use the queue fully disables it. The parameters TRIGGERLEVEL, OPENSTATUS, and ENABLESTATUS can be altered regardless of whether the queue is enabled or disabled. The value of the ENABLESTATUS parameter cannot be altered while a queue is in a “disable pending” state.

A transient data queue cannot be disabled while it is in use, or while tasks are waiting to use it.

Indirect and remote queues can be disabled at any time because they have no concept of being “in use”.

If tasks are waiting to use an extrapartition queue, a physically recoverable queue, or a non-recoverable intrapartition queue, and an attempt is made to disable the queue, the queue enters a “disable pending” state. The last task to use the extrapartition queue fully disables it.

If an attempt is made to disable a logically recoverable intrapartition TD queue when there are UOWs enqueued upon it, the queue enters a “disable pending” state. The last UOW to obtain the enqueue fully disables the queue. If a UOW has updated a logically recoverable queue and suffers an indoubt failure, the queue cannot be disabled until the indoubt failure has been resolved.

If a UOW owns an enqueue on a queue that is in a “disable pending” state, it is allowed to continue making updates.

When a queue is in a “disable pending” state, no new tasks can alter the queue’s state or its contents. A disabled response is returned when a READQ, WRITEQ, or DELETEQ request is issued against a destination that is in a “disable pending” state.

Note: If a task updates a logically recoverable transient data queue, and attempts to disable the queue and alter an attribute of the queue (for example, ATITRANID) within the same UOW, the call fails. This is because the UOW is a user of the queue, and the queue enters a “disable pending” state. The SET operation on the queue attribute, in this case, ATITRANID, fails. The queue does not become fully disabled until the UOW commits or backs out at syncpoint. You are recommended

to issue an EXEC CICS SYNCPOINT command before attempting to update the queue attribute (ATITRANID) using SET TDQUEUE.

Options

ATIFACILITY(*cvda*) (intrapartition queues only)

specifies whether the queue has a terminal (or session) associated with it.

When ATI occurs, this option determines whether the task that CICS creates to process the queue has a principal facility or not. CVDA values are:

NOTERMINAL

ATI tasks are to execute without a principal facility.

TERMINAL

ATI tasks require the terminal specified in ATITERMID as the principal facility.

ATITERMID(*data-value*) (intrapartition queues only)

specifies the 4-character name of the terminal or session associated with the queue, if any. When CICS creates a task to process the queue, this terminal is the principal facility if the ATIFACILITY value is TERMINAL.

You can set this value at any time, but it is used only during ATI, and only when ATI tasks are to have a principal facility. When ATIFACILITY is NOTERMINAL, CICS retains but does not use the ATITERMID value, and does not display it in an INQUIRE TDQUEUE command.

ATITRANID(*data-value*) (intrapartition queues only)

specifies the 4-character identifier of the transaction to be executed when CICS initiates a task automatically to process the queue. This value is used only during ATI. CICS does not check the ATITRANID value when you set it but, when ATI occurs, the created task abends unless the ATITRANID value names a transaction defined at the time. Furthermore, this transaction must not be defined as remote.

ATIUSERID(*data-value*) (intrapartition queues only)

specifies the 8-byte user identifier associated with the queue, if any. If there is no terminal associated with the queue when ATI occurs, CICS assigns this user to the task it creates to process the queue.

You can set this value at any time, but it is used only during ATI, and only when the ATIFACILITY value is NOTERMINAL. When ATIFACILITY is TERMINAL, CICS retains but does not use the ATIUSERID value, and does not display it in an INQUIRE TDQUEUE command.

In addition to the authority checks made for any SET TDQUEUE command, when ATIUSERID is specified, CICS invokes the external security manager to ensure that the user associated with the task issuing the command has authority to act for the user named in ATIUSERID. When the ESM is RACF, this means that the user associated with the task must be defined as a RACF **surrogate** for the user in ATIUSERID.

ENABLESTATUS(*cvda*)

specifies whether the queue can be accessed by applications. CVDA values are:

DISABLED

The queue cannot be accessed by applications. You cannot disable a queue that has suffered an indoubt failure.

ENABLED

The queue can be accessed by applications.

For extrapartition queues, changing the ENABLESTATUS value affects only the availability of the queue; CICS does not open or close the associated data set.

OPENSTATUS(*cvda*) (extrapartition queues only)

specifies whether the data set associated with the queue is to be open or closed. CVDA values are:

CLOSED

The data set is to be closed.

OPEN The data set is to be opened.

TDQUEUE(*data-value*)

specifies the 4-character name of the transient data queue whose attributes you are changing.

TRIGGERLEVEL(*data-value*) (intrapartition only)

specifies, as a fullword binary value, the number of items that must be on the queue for ATI to occur, or, alternatively, that ATI is not to occur. The number must be in the range 0–32767. If it is zero, ATI does not occur. If it is not zero, when the queue reaches the TRIGGERLEVEL depth CICS creates a task to process it automatically. See also the ATIFACILITY, ATITERMINAL, ATITRANSID, and ATIUSERID options.

Conditions

INVREQ

RESP2 values:

- 2 TRIGGERLEVEL was specified for an extrapartition queue.
- 3 The TRIGGERLEVEL value is not in the range 0–32767.
- 4 ATITERMID was specified for an extrapartition queue.
- 5 ATITRANID was specified for an extrapartition queue.
- 6 ATIFACILITY was specified for an extrapartition queue.
- 7 ATIFACILITY has an invalid CVDA value.
- 8 OPENSTATUS has an invalid CVDA value.
- 9 OPENSTATUS was specified for an intrapartition queue.
- 10 ENABLESTATUS has an invalid CVDA value.
- 12 The queue is remote.
- 13 The queue is indirect.
- 16 OPENSTATUS was specified, but the JCL DDNAME to which the queue definition points was not found.
- 18 SET not possible because the queue was not closed.
- 19 ATIUSERID was specified for an extrapartition queue.
- 20 The ESM interface is not initialized.
- 21 CICS has received an unknown response from the ESM.
- 22 The ESM did not respond.
- 30 Disabled pending condition.
- 31 SET not possible because the queue was not disabled.
- 35 SET not possible because the queue is indoubt.
- 40 SET not possible because the queue is CXRF.

IOERR

RESP2 values:

- 14 An error occurred opening or closing the data set associated with the queue.
- 17 The queue cannot be set CLOSED because there is no space in the associated data set.

NOTAUTH

RESP2 values:

- 23 The user named on the ATIUSERID option is not authorized.
- 24 The user named in ATIUSERID has been revoked.
- 25 During SECLABEL processing by the external security manager, an error occurred. For information about security labels, see the *RACF Security Administrator's Guide*.
- 27 The user named in the ATIUSERID option is not allowed to access the queue.
- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.
- 102 The user associated with the issuing task is not an authorized surrogate for the user specified in ATIUSERID.

QIDERR

RESP2 values:

- 1 The queue cannot be found.

USERIDERR

RESP2 values:

- 28 The user named in ATIUSERID is not known to the ESM.

SET TEMPSTORAGE

Set the amount of storage that is available to temporary storage queues in the CICS region.

SET TEMPSTORAGE

►►—SET TEMPSTORAGE—TSMAINLIMIT(*data-value*)—————►►

Conditions: INVREQ, NOTAUTH

Description

The **SET TEMPSTORAGE** command changes the limit for the amount of storage that is available for main temporary storage queues to use.

When you change this limit, check your current setting for the z/OS parameter **MEMLIMIT**. **MEMLIMIT** limits the amount of 64-bit storage that the CICS address space can use. Your setting for **TSMAINLIMIT** must not be greater than 25% of the **MEMLIMIT** value. Use the CICS SPI command **INQUIRE SYSTEM MEMLIMIT** to find the **MEMLIMIT** value that currently applies to the CICS system.

Options

TSMAINLIMIT(*data-value*)

Specifies a doubleword binary value for the maximum amount of storage that CICS makes available for main temporary storage queues to use. The

minimum value is 1048576 bytes (1 MB), and the maximum value is 34359738368 bytes (32,768 MB or 32 GB).

A value entered in bytes is rounded down to the nearest megabyte.

If you increase the TSMAINLIMIT setting, the value is set as follows:

- If the new value is not greater than 25% of the value of the z/OS parameter **MEMLIMIT**, the value that you choose is set.
- If the new value is greater than 25% of the **MEMLIMIT** value, TSMAINLIMIT remains unchanged.

If you decrease the TSMAINLIMIT setting, CICS attempts to maintain at least 25% free space in allowed storage above current utilization, so that temporary storage write requests do not reach TSMAINLIMIT too rapidly. The value is set as follows:

- If there is currently less than 25% free space, TSMAINLIMIT remains unchanged.
- If at least 25% of the new limit will be free space, the setting is decreased to the value that you choose.
- If less than 25% of the new limit would be free space, the setting is decreased to the current utilization plus 33% of that utilization.

Conditions

INVREQ

RESP2 value:

- 1 An attempt was made to set an invalid value.
- 2 TSMAINLIMIT must not be greater than 25% of MEMLIMIT.

NOTAUTH

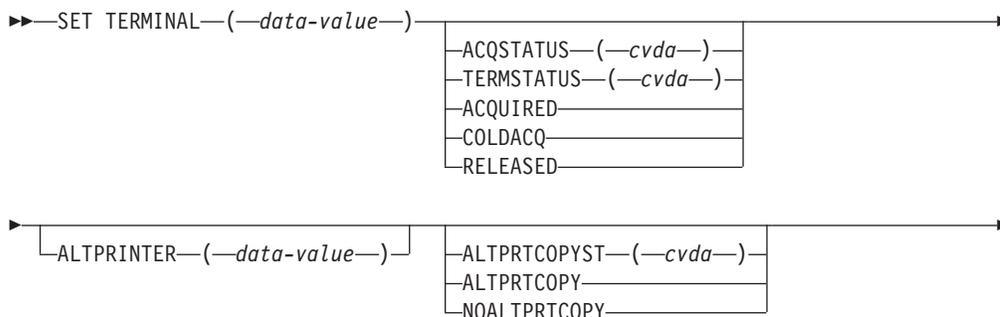
RESP2 value:

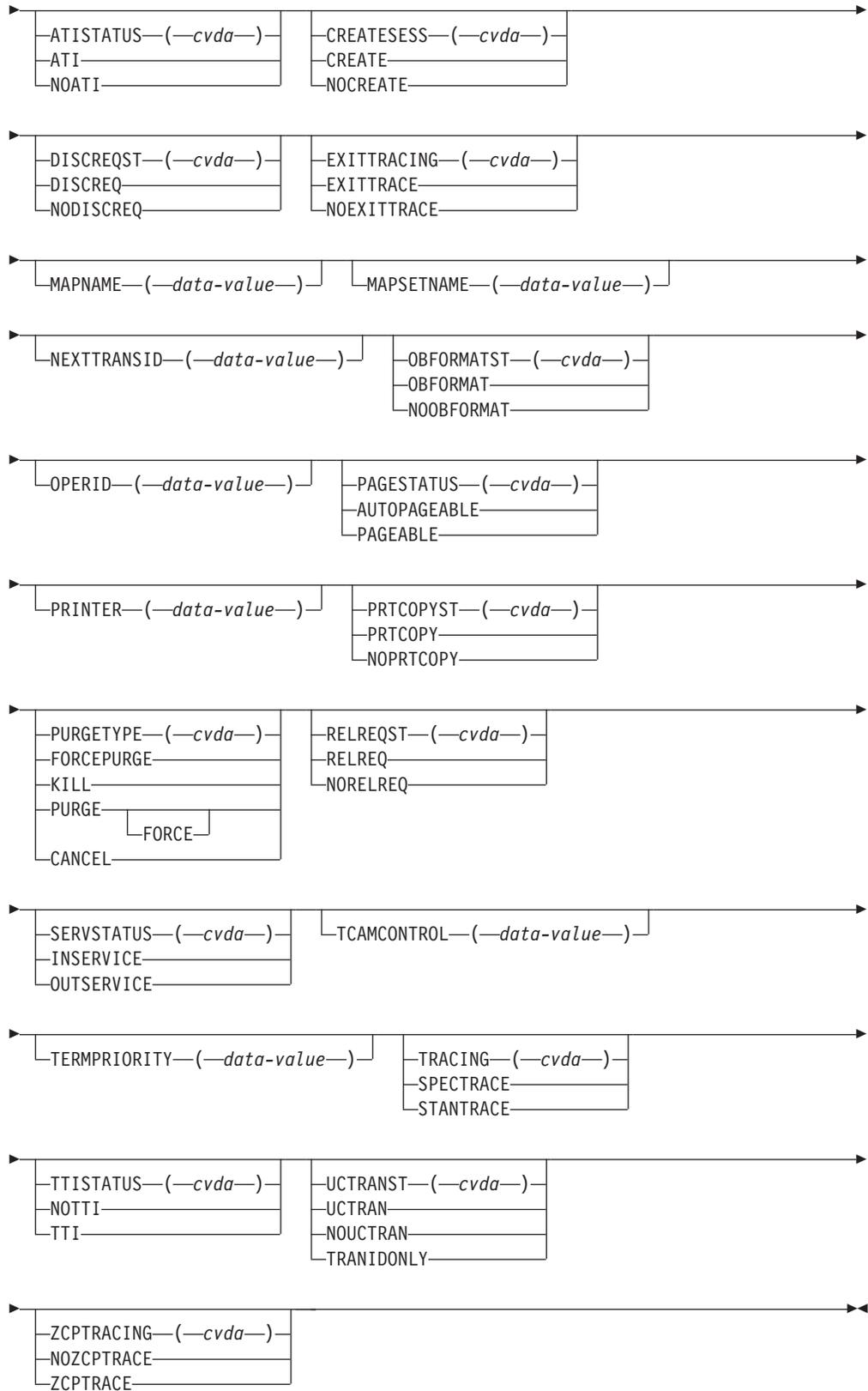
- 100 The user associated with the issuing task is not authorized to use this command.

SET TERMINAL

Change some terminal attributes and cancel outstanding AIDs.

SET TERMINAL





Conditions: INVREQ, NORMAL, NOTAUTH, TERMIDERR

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The SET TERMINAL command changes some of the values of a named terminal definition. It cannot be used for APPC sessions.

Only PURGETYPE, PURGE, or FORCEPURGE can be used for IRC sessions.

If a terminal TCTTE is available in a remote system, in either model or surrogate form, a change can be made to TRACING or NEXTTRANSID in the remote definition. This change is not shipped back to the TOR. This allows the user to make a change that applies only to the remote TCTTE.

The SET TERMINAL command can also be used to change the UCTRANST option of a surrogate terminal. This change is shipped back to the TOR and intermediate systems. Any attempt to change any other attribute for a model or surrogate terminal results in INVREQ with RESP2=24.

Options

ACQSTATUS(*cvda*) (z/OS Communications Server only)

This option is retained only for compatibility purposes. You should use the TERMSTATUS option in new applications.

ALTPRINTER(*data-value*)

Specifies the name of a 3270 printer for use as an alternative to the printer defined on the PRINTER option. The name can be up to four characters long.

Note: You cannot specify ALTPRINTER for a terminal that does not have a primary printer defined (on the PRINTER parameter).

See the *CICS Resource Definition Guide* for information about the PRINTER and ALTPRINTER parameters for defining primary and alternate printers for terminals.

Note: For z/OS Communications Server terminals, in a transaction routing environment, this command does not take effect until the next flow across the link from the TOR to the AOR for the named terminal.

ALTPRTCOPYST(*cvda*)

Specifies the alternate printer copy status. This indicates whether CICS is to use the hardware COPY feature to satisfy a print request on the printer named on the ALTPRINTER parameter. CVDA values are:

ALTPRTCOPY

CICS is to use the hardware COPY feature to satisfy a print request on the alternate printer.

NOALTPRTCOPY

CICS is not to use the hardware COPY feature.

Note: You cannot specify ALTPRTCOPY for a terminal that does not have an alternate printer defined.

See the *CICS Resource Definition Guide* for information about the ALTPRINTCOPY parameter, which specifies the use of the hardware copy feature for the alternate printer on the terminal definition.

ATISTATUS (*cvda*)

Specifies whether the terminal can be used by transactions that are automatically initiated from within CICS or, if the terminal is an ISC session, by transactions that are using this session as an alternate facility to communicate with another system. CVDA values are:

ATI The terminal can be used by automatically initiated transactions.

NOATI

The terminal cannot be used by automatically initiated transactions.

A terminal cannot have both NOATI and NOTTI in its status.

CREATESESS (*cvda*) (**z/OS Communications Server only**)

Specifies whether the terminal can be acquired automatically by ATI transactions. CVDA values are:

CREATE

The terminal can be acquired automatically.

NOCREATE

The terminal cannot be acquired automatically.

DISCREQST (*cvda*)

Specifies whether CICS is to honor a disconnect request from the terminal. CVDA values are:

DISCREQ

CICS honors a disconnect request for a z/OS Communications Server device, and issues a z/OS Communications Server CLSDST macro instruction to terminate the z/OS Communications Server session with that logical unit.

It also means that CESF LOGOFF (or GOODNIGHT) from the terminal causes disconnection.

NODISCREQ

CICS does not honor a disconnect request for a z/OS Communications Server device.

EXITTRACING (*cvda*)

Specifies whether the activity associated with the terminal exit program is to be traced. CVDA values are:

EXITTRACE

Exit program activity is to be traced.

NOEXITTRACE

Exit program activity is not to be traced.

MAPNAME (*data-area*)

Specifies the 7-character name of the map that is to be saved (stored) by CICS as the name of the last map sent to this device. If this terminal is a surrogate, the map name specified is returned in the DETACH sequence to the terminal-owning region when the currently executing transaction terminates, unless the map name is superseded by a subsequent SEND MAP command. You can use the MAPNAME option to restore a map name that was returned to the application program in a previous INQUIRE TERMINAL command. If the terminal is not supported by BMS (for example, this terminal is a session), an INVREQ condition is raised with a RESP2 value of 60.

MAPSETNAME (*data-area*)

Specifies the 8-character name of the mapset that is to be saved by CICS as the

name of the last mapset used in a SEND MAP command processed for this terminal. If this terminal is a surrogate, the mapset name specified is returned in the DETACH sequence to the terminal-owning region when the currently executing transaction terminates, unless the mapset name is superseded by a subsequent SEND MAP command. The MAPSETNAME option can be used to restore a mapset name that was returned to the application program in a previous INQUIRE TERMINAL command. If the terminal is not supported by BMS (for example, this terminal is a session), an INVREQ condition is raised with a RESP2 value of 60.

NEXTTRANSID(*data-value*)

Specifies the next transaction identifier for the specified terminal. The identifier can be up to 4 characters long. If you specify the NEXTTRANSID parameter as blanks (X'40404040'), CICS sets the next transaction identifier to nulls, meaning there is no NEXTTRANSID defined for the terminal.

Changes are permitted to a remote TCTTE, but the change is not shipped back to the TOR.

Note: NEXTTRANSID cannot be set if a transaction has been defined for this terminal.

OBFORMATST(*cvda*)

Specifies whether the device supports outboard formatting. See the TYPETERM definition in the *CICS Resource Definition Guide* for details of the types of device that support outboard formatting. CVDA values are:

NOOBFORMAT

The device does not support outboard formatting.

OBFORMAT

The device supports outboard formatting.

Note: OBFORMATST cannot be specified for a console or 3790.

OPERID(*data-value*)

Specifies an operator identification code that is to be associated with the terminal. The identification code can be up to 3 characters long. The operator identification code will continue to be associated with the terminal until it is changed by another SET TERMINAL OPERID command, or until the user signed on at the terminal changes (i.e. until a user signs on or signs off at the terminal).

PAGESTATUS(*cvda*)

Specifies how pages are to be written. CVDA values are:

AUTOPAGEABLE

Pages, after the first in a series, are to be written to the terminal automatically.

PAGEABLE

Pages, after the first in a series, are to be written to the terminal on request from the operator.

PRINTER(*data-value*)

Specifies the name of the primary printer CICS is to use in response to a print request (either an ISSUE PRINT command, or a PRINT request from an operator pressing a program access (PA) key). The name can be up to 4 characters long. See the *CICS Resource Definition Guide* for information about specifying 3270-type printers.

Note: For z/OS Communications Server terminals, in a transaction routing environment, this command does not take effect until the next flow across the link from the TOR to the AOR for the named terminal.

PRTCOPYST(*cvda*)

Specifies whether CICS is to use the hardware COPY feature to satisfy a print request on the printer named on the PRINTER parameter. CVDA values are:

NOPTCOPY

CICS is not to use the hardware COPY feature.

PRTCOPY

CICS is to use the hardware COPY feature to satisfy a print request on the primary printer.

Note: You cannot specify PRTCOPY for a terminal that does not have a printer defined.

See the *CICS Resource Definition Guide* for information about the PRINTCOPY parameter, which specifies the use of the hardware copy feature for the primary printer on the terminal definition.

PURGETYPE(*cvda*)

Specifies whether transactions running with the named terminal can be purged. CVDA values are:

CANCEL

AIDs queuing for the specified terminal are to be canceled. AIDs representing scheduled and allocated requests waiting in the local CICS system for the specified terminal are canceled. However, CRSR AIDs and TD AIDs with an associated triggered task already started are not canceled.

When a canceled scheduled request is found to have a precursor in a remote CICS system, this remote AID is canceled asynchronously. Message DFHTF0100 is written to CSMT to indicate how many AIDs have been deleted for the terminal and how many remain.

FORCEPURGE

Transactions are to be purged immediately. This can lead to unpredictable results and should be used only in exceptional circumstances.

Kill

The task is to be terminated. System and data integrity is not guaranteed. The KILL option extends the PURGE and FORCEPURGE options. It should be used only after an attempt has been made to PURGE or FORCEPURGE a task. The KILL option does not guarantee integrity of any kind but in some situations it allows the user to free up a stalled region enabling the region to continue processing. In some cases, for example, if a task is killed during backout processing, CICS terminates abnormally.

PURGE

The transactions can be terminated only if system and data integrity can be maintained. A transaction is not to be purged if its definition specifies SPURGE=NO.

FORCEPURGE replaces PURGE FORCE, which is retained only for compatibility purposes. You should use FORCEPURGE in new applications.

PURGETYPE cannot be specified for non-z/OS Communications Server terminals.

RELREQST (*cvda*)

Specifies the status for releasing the logical unit. CVDA values are:

NORELREQ

CICS is not to release the logical unit upon request by another z/OS Communications Server application program.

RELREQ

CICS is to release the logical unit, if the logical unit is not currently busy running a transaction.

SERVSTATUS (*cvda*)

Specifies whether the terminal is to be in- or out-of-service. CVDA values are:

INSERVICE

CICS is to set the terminal in-service and available for use.

OUTSERVICE

CICS is to set the terminal out-of-service, and not available for transactions. Unless you specify PURGE or FORCEPURGE, any current transaction is allowed to terminate normally, but no further transactions are allowed to use the terminal.

If the execution diagnostic facility (EDF) is in use at the specified terminal, EDF stops immediately, because it is a sequence of separate transactions, while the transaction that is being tested under EDF is allowed to complete.

If you set a z/OS Communications Server terminal to OUTSERVICE, it is also RELEASED and the operator is signed off, either immediately or when the current transaction has terminated. You cannot therefore set the terminal associated with the executing transaction to OUTSERVICE, unless it is a printer.

TCAMCONTROL (*data-value*) (**TCAM/DCB remote terminals only**)

Obsolete. TCAM terminals are not supported.

TERMINAL (*data-value*)

Specifies the 4-character terminal name.

Note: As a result of the operation of the XICTENF and XALTENF global user exits, it is possible for SCHEDULE requests to be queued for a terminal that is not yet defined to the local CICS system. You can use the SET TERMINAL(*data-value*) CANCEL command to remove these requests.

TERMPRIORITY (*data-value*)

Specifies, as a fullword binary value, the priority required for the terminal, relative to other terminals, in the range 0–255.

TERMSTATUS (*cvda*) (**z/OS Communications Server only**)

Specifies the session status for the logical unit represented by this terminal. CVDA values are:

ACQUIRED

CICS is to acquire a session with the logical unit represented by this terminal.

COLDACQ

CICS is to acquire a session with the logical unit represented by this terminal where no resynchronization is required.

RELEASED

CICS is to terminate the session. This happens immediately if you also

specify the PURGE option, otherwise the session is terminated when the current active transaction finishes.

TRACING(*cvda*)

Specifies the required tracing activity associated with the terminal. CVDA values are:

SPECTRACE

Special tracing is to be used.

STANTRACE

Standard tracing is to be used.

Changes are permitted to a remote TCTTE, but the change is not shipped back to the TOR.

TTISTATUS(*cvda*)

Specifies whether this terminal can be used by the transactions that are initiated from this terminal. CVDA values are:

NOTTI

This terminal cannot be used by transactions initiated from it.

TTI This terminal can be used by transactions initiated from it.

A terminal cannot be defined with both NOATI and NOTTI.

UCTRANST(*cvda*)

Specifies whether the uppercase translate option is to be set for transactions associated with this terminal. Note that there is also an UCTRAN option on the profile definition. See Table 7 on page 741 for information on how the UCTRAN options on the terminal and transaction profiles interact.

If a terminal TCTTE is available in a remote system, in either model or surrogate form, a change can be made to TRACING or NEXTTRANSID in the remote definition. This change is not shipped back to the TOR. This allows the user to make a change which applies only to the remote TCTTE. The SET TERMINAL command can also be used to change the UCTRANST option of a surrogate terminal. This change is shipped back to the TOR and intermediate systems. Attempting to change any other attribute for a model or surrogate terminal results in INVREQ with RESP2=24.

This command may be used to set the uppercase translation option for a remote terminal, if the named terminal is the principal facility of the task issuing the command. If the remote terminal is not the principal facility, the INVREQ condition is raised with a RESP2 value of 24. The uppercase translation option is also changed in the terminal-owning region and any intermediate region in a daisy-chaining setup. CVDA values are:

NOUCTRAN

CICS is not to perform uppercase translation on input from this terminal (unless specified otherwise on the profile for individual transactions).

TRANIDONLY

CICS is to perform uppercase translation on the transaction id only on input from this terminal.

UCTRAN

CICS is to perform uppercase translation on input from this terminal.

Table 7. The effect of the UCTRAN parameters

Profile	Terminal (TYPETERM)		
	UCTRAN (YES)	UCTRAN (NO)	UCTRAN (TRANID)
UCTRAN (YES)	Tranid: Yes Data: Yes	Tranid: No Data: Yes	Tranid: Yes Data: Yes
UCTRAN (NO)	Tranid: Yes Data: Yes	Tranid: No Data: No	Tranid: Yes Data: No

Note: This table shows which portion of the terminal input is translated (transaction id and/or data) according to the setting of the UCTRAN on the PROFILE and TYPETERM resource definitions.

ZCPTRACING(*cvda*)

Specifies the required tracing activity associated with the z/OS Communications Server control component of CICS. CVDA values are:

NOZCPTRACE

z/OS Communications Server ZCP tracing is not to be carried out.

ZCPTRACE

z/OS Communications Server ZCP tracing is to be carried out.

Conditions

INVREQ

RESP2 values:

- 1 TERMSTATUS or ACQSTATUS was specified for an IRC session or non-z/OS Communications Server terminal.
- 2 TERMSTATUS or ACQSTATUS has an invalid CVDA value.
- 4 ATISTATUS has an invalid CVDA value.
- 5 ATISTATUS change would result in NOATI and NOTTI.
- 6 CREATESESS was specified for non-z/OS Communications Server terminal.
- 7 CREATESESS has an invalid CVDA value.
- 9 PAGESTATUS has an invalid CVDA value.
- 11 Trying to put the issuing terminal OUTSERVICE.
- 13 SERVSTATUS has an invalid CVDA value.
- 15 TERMPRIORITY value not in range 0–255.
- 17 NOTTI cannot be specified for the issuing terminal.
- 18 TTISTATUS has an invalid CVDA value.
- 21 PURGETYPE has an invalid CVDA value.
- 22 TRACING has an invalid CVDA value.
- 24 Invalid option requested for a remote terminal.
- 25 ACQUIRED specified, but terminal is not in service.
- 26 PURGE specified, but target task has SPURGE=NO on its associated transaction definition.
- 27 EXITTRACING has an invalid CVDA value.
- 28 ZCPTRACING has an invalid CVDA value.
- 29 EXITTRACING or ZCPTRACING specified for a non-z/OS Communications Server terminal (or z/OS Communications Server not installed).
- 31 This is a remote terminal with no associated surrogate.
- 33 SET TERMINAL is not valid for an LU6.2 (APPC) session.
- 34 A permanent transaction has been defined for this terminal (TRANSACTION operand in TERMINAL definition).
- 35 Attempt made to change TCAM CONTROL on non-TCAM terminal.
- 36 Invalid value supplied for TCAM CONTROL.
- 37 Preset signon failed, terminal remains OUTSERVICE.

- 38 OBFORMATST has an invalid CVDA value.
- 39 RELREQST has an invalid CVDA value.
- 40 DISCREQST has an invalid CVDA value.
- 41 ALTPRTPCOPIST has an invalid CVDA value.
- 42 PRTPCOPIST has an invalid CVDA value.
- 43 UCTRANST has an invalid CVDA value.
- 44 Options would result in the invalid combination of the alternate printer copy status being set without an alternate printer defined.
- 45 Options would result in the invalid combination of the alternate printer being defined without a primary printer defined.
- 46 OBFORMATST is specified for a console or 3790.
- 48 Options would result in the invalid combination of the printer copy status being set without a primary printer defined.
- 50 z/OS Communications Server not available for z/OS Communications Server terminal.
- 51 PRINTER and ALTPRINTER option specified for a terminal that is not z/OS Communications Server 3270 or 3270 compatibility mode.
- 52 PRTPCOPIST or ALTPRTPCOPST option specified for a terminal that is not z/OS Communications Server 3270 or 3270 compatibility mode.
- 54 Option other than PURGETYPE specified for IRC session.
- 57 Other SET parameters were included with the CANCEL option.
- 61 No previous attempt has been made to forcepurge the task whose facility is a terminal.

NORMAL

RESP2 values:

- 53 Purge deferred.
- 58 AIDs are successfully canceled.
- 59 No AIDs are canceled.
- 60 MAPNAME or MAPSETNAME specified, but the terminal is not of a type supported by BMS.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

TERMIDERR

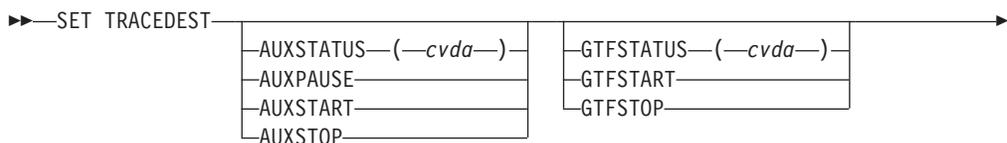
RESP2 values:

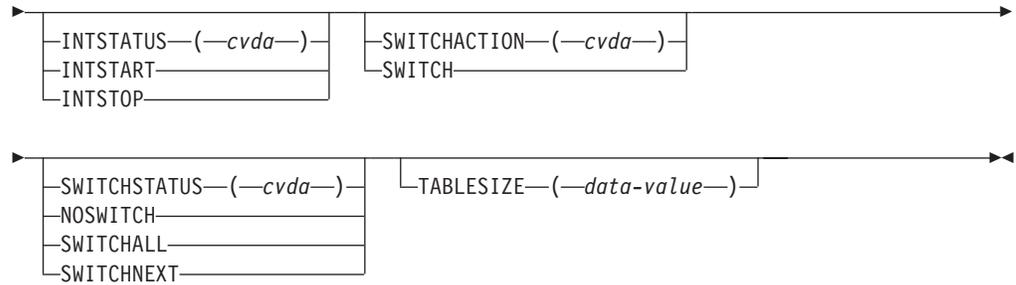
- 23 The named terminal cannot be found.

SET TRACEDEST

Change tracing options.

SET TRACEDEST





Conditions: INVREQ, IOERR, NOSPAC, NOSTG, NOTAUTH

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

CICS can write trace entries to three possible destinations: the CICS internal trace table, the auxiliary trace data set, and the MVS Generalized Trace Facility (GTF). You can use the SET TRACEDEST command to specify which destinations receive trace entries. You also can use it to change the size of the trace table and to switch auxiliary trace data sets.

Two other commands, SET TRACEFLAG and SET TRACETYPE, and a CICS-supplied transaction, CETR, can be used to control the number and type of trace entries.

Changes made with this command are not recorded in the CICS catalog. Therefore the options affected are always reset to the corresponding system initialization values at CICS startup. These are TRTABSZ (for internal tracing), AUXTR and AUXTRSW (auxiliary tracing), and GTFTR (GTF tracing).

Options

AUXSTATUS(*cvda*)

Specifies whether auxiliary tracing takes place; that is, whether trace entries are written to the active CICS auxiliary trace data set. (See the SWITCHACTION option for more about auxiliary trace data sets.) CVDA values are as follows:

AUXPAUSE

CICS stops writing entries, but leaves the data set open at its current position. A subsequent AUXSTART request will resume writing entries immediately after those that preceded the AUXPAUSE request. You can specify AUXPAUSE only when auxiliary tracing is currently active.

AUXSTART

CICS starts writing entries. The data set is opened first if it is currently closed.

AUXSTOP

CICS stops writing entries. The data set is closed if it is open. A subsequent AUXSTART request causes CICS to write new entries at the start of the data set, overwriting the previous contents, unless there are two auxiliary trace data sets and they are switched between the AUXPAUSE and AUXSTART.

GTFSTATUS (*cvda*)

Specifies whether trace entries are sent to the MVS Generalized Tracing Facility (GTF). CVDA values are as follows:

GTFSTART

Entries are sent.

GTFSTOP

Entries are not sent.

Note: A value of GTFSTART is necessary but not sufficient for recording CICS trace entries on GTF. In addition, CICS must be initialized with GTF support (the GTFTR system initialization option), and GTF must be started in MVS with the TRACE=USR option.

INTSTATUS (*cvda*)

Specifies whether internal tracing occurs; that is, whether non-exception trace entries are recorded in the internal trace table. (Exception entries are always recorded.) CVDA values are as follows:

INTSTART

Entries are recorded.

INTSTOP

Entries are not recorded.

SWITCHACTION (*cvda*)

Specifies that CICS must switch the auxiliary trace data sets.

If your system supports auxiliary tracing, it has either one or two auxiliary trace data sets. One is active, which means it receives trace entries when auxiliary tracing is turned on, and the other, if there are two, is a standby.

When there are two, you can reverse their roles by specifying SWITCH. This causes CICS to close the current active data set, open the standby, and reverse the designation of which is active and which is a standby.

If there is only one auxiliary trace data set (or none), SWITCH causes an exception condition, because CICS attempts to open a data set that is not defined.

The CVDA value is as follows:

SWITCH

CICS performs a switch.

Note: If you request AUXSTATUS and SWITCHACTION in the same command, AUXSTATUS is set first.

SWITCHSTATUS (*cvda*)

Specifies the action CICS takes when the current active auxiliary trace data set fills. When this occurs, CICS cannot continue auxiliary tracing unless a switch or an AUXSTOP-AUXSTART sequence takes place (see the SWITCHACTION and AUXSTATUS options). CVDA values are as follows:

NOSWITCH

CICS takes no action.

SWITCHALL

CICS switches every time the active data set fills.

SWITCHNEXT

CICS switches when the current data set is full, but only once; thereafter NOSWITCH is in effect.

TABLESIZE(*data-value*)

Specifies, as a fullword binary value, the size of the internal trace table in kilobytes. If you specify a value that is different from the current trace table size, CICS suspends internal tracing while the change is made, and data in the old table is deleted.

The table is allocated in multiples of 4 KB, with a minimum size of 16 KB. If you specify a value that is not a multiple of the page size (4 KB), it is rounded up to the next multiple of 4 KB. If you specify less than 16 KB, the value is rounded up to 16 KB. The maximum size is 1048576 KB (1 GB).

CICS can use 64-bit (above-the-bar) storage rather than 31-bit (above-the-line) storage for the internal trace table, depending on the version of the z/OS operating system and whether the CICS region operates with transaction isolation. See CICS facilities that can use 64-bit storage in the Performance Guide.

- When CICS uses 31-bit storage for the internal trace table, if you increase the value of TABLESIZE remember that there must be enough MVS page storage above the line for both the internal trace table and the EDSAs. To check the relevant storage limit that currently applies to the CICS region, and for more information about the z/OS **REGION** parameter, see Estimating and setting the REGION parameter in the CICS Performance Guide.
- When CICS uses 64-bit storage for the internal trace table, the value of TABLESIZE must be less than the value of the z/OS **MEMLIMIT** parameter, and you must also allow for other facilities in the CICS region that use 64-bit storage. See Estimating, checking, and setting MEMLIMIT in the Performance Guide.

Conditions

INVREQ

RESP2 values:

- 1 INTSTATUS has an invalid CVDA value.
- 2 A TABLESIZE value of < -1 has been specified.
- 3 AUXSTATUS has an invalid CVDA value.
- 4 SWITCHSTATUS has an invalid CVDA value.
- 5 GTFSTATUS has an invalid CVDA value.
- 6 AUXPAUSE was specified, but auxiliary tracing is not active.
- 11 SWITCHACTION has an invalid CVDA value.

IOERR

RESP2 values:

- 10 A SWITCH request or a SET AUXSTART request resulted in an “open” error for the trace data set.

NOSPACE

RESP2 values:

- 7 There is insufficient space for the new trace table.

NOSTG

RESP2 values:

- 8 There is insufficient space for an auxiliary trace buffer.
- 9 There is insufficient space for a GTF trace buffer.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

Examples

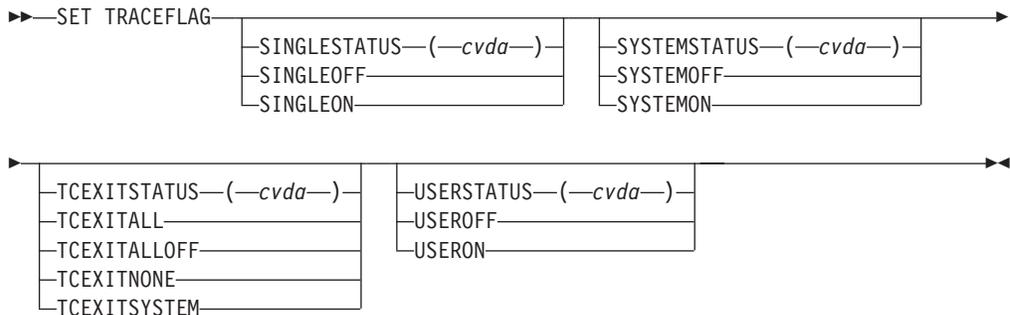
```
EXEC CICS SET TRACEDEST
        SWITCH
        NOSWITCH
```

The SWITCH option tells CICS to switch now from the active auxiliary trace data set (which is not necessarily full) to the alternate auxiliary trace data set. The NOSWITCH option tells CICS not to switch when the new active data set fills.

SET TRACEFLAG

Change settings of trace flags.

SET TRACEFLAG



Conditions: INVREQ, NOTAUTH

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The SET TRACEFLAG command allows you to change the flags that control the creation of trace entries in CICS. (See the *CICS Problem Determination Guide* for more information about tracing facilities and control.)

Changes made with this command are not recorded in the CICS catalog, and therefore do not persist beyond CICS shutdown.

Options

SINGLESTATUS(*cvda*)

specifies whether tracing is to be turned on or suppressed for the task issuing

this SET TRACEFLAG command. No nonexception trace entries are made for a task when this flag is off (exception trace entries are *always* recorded).

When tracing is allowed, the type of tracing is standard unless special tracing has been requested (in an earlier use of the CETR transaction) for the transaction being executed or the terminal that is the principal facility. CVDA values are:

SINGLEOFF

Tracing is suppressed.

SINGLEON

Tracing is allowed.

SYSTEMSTATUS(*cvda*)

specifies how the system master trace flag is to be set. This flag determines whether CICS makes or suppresses standard trace entries (it does not govern special or exception trace entries). It applies to all tasks and all system activity; however, for standard trace entries to be recorded for any particular task, both the system master flag and the SINGLESTATUS flag for the task must be on. CVDA values are:

SYSTEMOFF

Standard tracing is to be suppressed.

SYSTEMON

Standard tracing is to be active.

TCEXITSTATUS(*cvda*) (z/OS Communications Server only)

specifies which invocations of the CICS z/OS Communications Server exits are to be traced.

Two types of exit activity can be traced: invocations associated with particular terminals that have been designated for z/OS Communications Server exit tracing (“terminal-specific” activity), and invocations not related to any particular terminal (“nonterminal-specific” activity). You can trace both types or nonterminal-specific activity only.

CVDA values are:

TCEXITALL

All exit activity is to be traced.

TCEXITALLOFF

Terminal-specific activity is not to be traced. The status of nonterminal-specific tracing is to remain unchanged.

TCEXITNONE

No exit activity is to be traced.

TCEXITSYSTEM

Nonterminal-specific activity is to be traced, but terminal-specific activity is not.

USERSTATUS(*cvda*)

specifies whether the user master trace flag is to be set on or off. This flag governs whether non-exception user trace entries are recorded or suppressed (entries that specify the EXCEPTION option are never suppressed). It applies to all tasks; however, for user entries to be recorded for any particular task, both the user master trace flag and the SINGLESTATUS flag for that task must be on. CVDA values are:

USEROFF
User tracing is suppressed.

USERON
User tracing is allowed.

Conditions

INVREQ

RESP2 values:

- 1 SYSTEMSTATUS has an invalid CVDA value.
- 2 USERSTATUS has an invalid CVDA value.
- 3 SINGLESTATUS has an invalid CVDA value.
- 4 TCEXITSTATUS has an invalid CVDA value.
- 5 TCEXITSTATUS is specified but z/OS Communications Server is not installed.

NOTAUTH

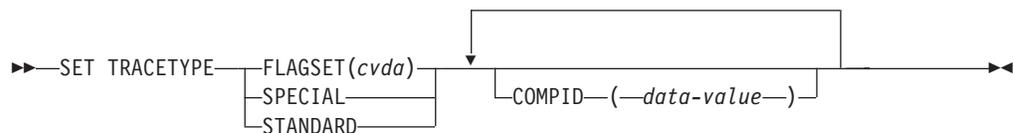
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SET TRACETYPE

Change the tracing levels of CICS components.

SET TRACETYPE



Conditions: INVREQ, NOTAUTH, NOTFND

Description

Use the SET TRACETYPE command to change the levels of tracing for one or more CICS components.

Each CICS component has trace levels defined separately for standard CICS tracing and special tracing. (See the *CICS Problem Determination Guide* for definitions of these terms and for information about CICS tracing in general.) You can set either type for any number of components in a SET TRACETYPE command, but you can set only one type per command.

For each component that you specify, you define the trace levels as a bit string. The bits are read from left to right; that is, the first bit corresponds to trace level 1, the second to trace level 2, and so on. A value of 1 turns on the trace level; 0 turns it off. For example, 'X'C0000000' turns on trace levels 1 and 2 and turns off all others.

Although most components define only a few trace levels, you must provide a 32-bit (4 byte) data value. CICS ignores bits that do not correspond to trace levels, and thus it does not matter whether you specify 0 or 1 for them.

The SJ component uses higher trace levels. Trace levels 29 – 32 for the SJ component correspond to JVM Level 0 trace, JVM Level 1 trace, JVM Level 2 trace, and JVM User trace respectively. The CICS-supplied transaction CETR includes screens that provide a better way to set these trace levels and to change the JVM trace options associated with them.

Options

COMPID(*data-value*)

Sets the trace levels for the CICS component identified by COMPID, using the bits in the data value as described above.

CICS components can be identified by a 2 character designation or, in some cases, a descriptive keyword. For example, to set the trace levels for the storage manager component of CICS, you can specify either:

```
SET TRACETYPE SM(data-value)
```

or

```
SET TRACETYPE STORAGE(data-value)
```

The following list shows all the 2 character identifiers, and the keywords for those components that have them.

AP	APPLICATION	Application
BA	BUSAPPMGR	Business applications manager
BM*		Basic mapping support
BR*	BRIDGE	3270 Bridge
CP*	CPI	Common programming interface
DC*		Dump control
DD	DIRMGR	Directory manager
DH	DOCUMENT	Document handling
DM	DOMAINMGR	Domain manager
DP	DEBUGTOOL	Debugging Profiles domain
DS	DISPATCHER	Dispatch manager
DU	DUMP	Dump manager
EC*	EVENTCAPTURE	Event capture
EI*		EXEC interface
EJ	ENTJAVA	Enterprise Java domain
EM	EVENTMGR	Event manager
EP	EVENTPROC	Event processing domain
FC*		File control and DL/I
GC	GLOBALCATLG	CICS global catalog manager
IC*		Interval control
IE	IPECI	ECI over TCP/IP domain
II	IIOP	IIOP domain
IS*		Intersystem communication
KC*		Task control
KE	KERNEL	Kernel
LC	LOCALCATLG	CICS local catalog manager
LD	LOADER	Program load manager
LG	LOGGER	Log manager
LM	LOCKMGR	Lock manager
ME	MESSAGE	Message manager

ML		Markup language domain
MN	MONITOR	Monitoring manager
NQ	ENQUEUE	Enqueue domain
OT	OBJECTTRAN	Object Transaction Service (OTS) domain
PA	PARAMGR	Parameter manager
PC*		Program control
PG	PROGMGR	Program manager
PI	PIPEMGR	Pipeline manager domain
PT	PARTNER	Partner manager
RA*	RMIADAPTERS	Resource manager adapters
RI*	RMI	Resource manager interface (RMI)
RL	RESLIFEMGR	Resource life-cycle domain
RM	RECOVERY	Recovery manager
RS	REGIONSTAT	Region status
RX	RRS	Resource recovery services
RZ	REQUESTSTRM	Request streams domain
SC*		Storage control
SH	SCHEDULER	Scheduler services domain for BTS
SJ	SJVM	CICS JVM domain
SM	STORAGE	Storage manager
SO	SOCKETS	Sockets
ST	STATISTICS	Statistics manager
SZ*		Front-end programming interface
TC*		Terminal control
TD*		Transient data
TI	TIMER	Timer manager
TR	TRACE	Trace manager
TS	TEMPSTORAGE	Temporary storage
UE*		User exit interface
US	USER	User interface
WB	WEB	Web domain
WU	WEBRESTMGR	System Management RESTful API
W2	WEB2	Web 2.0 domain
XM	TRANMGR	Transaction manager
XS	SECURITY	Security manager

Components marked * are subcomponents of the AP domain, and the trace entries for these components are produced with a trace point ID of AP nnnn.

FLAGSET (*cvda*)

Indicates whether the standard or special flags, for the specified component, are to be set. CVDA values are as follows:

SPECIAL

Specifies that you want to set levels for special tracing, for the components listed.

STANDARD

Specifies that you want to set levels for standard tracing, for the components listed.

Conditions

INVREQ

RESP2 values:

- 1 An incorrect value was specified for FLAGSET.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values:

- 1 At least one CICS component was not accessible. Trace levels were set for the other components.

SET TRANCLASS

Set limits for a transaction class.

SET TRANCLASS



SET1

|—SET TRANCLASS—(*data-value*)—|

SET2

|—MAXACTIVE—(*data-value*)—|

SET3

|—PURGETHRESH—(*data-value*)—|

Conditions: INVREQ, NOTAUTH, TCIDERR

Description

The SET TRANCLASS command allows you to change the limits that govern tasks within a particular transaction class. These are the maximum number of tasks that can run concurrently (the MAXACTIVE value) and the maximum number that can queue awaiting initial dispatch (the PURGETHRESH value).

Options

MAXACTIVE(*data-value*)

specifies, as a fullword binary value, the largest number of tasks in the transaction class which can run concurrently. The value can be in the range 0-999.

Raising the MAXACTIVE limit has an immediate effect if the old value of MAXACTIVE has caused queuing, because CICS dispatches queued tasks up to the new MAXACTIVE value. The effect of lowering MAXACTIVE, however, is gradual. Tasks in the class that are already running are allowed to complete normally, but new tasks are not dispatched until the number running drops

below the new limit. If you lower MAXACTIVE to zero, you prevent any task in the class from starting execution until MAXACTIVE is increased.

PURGETHRESH(*data-value*)

specifies, as a fullword binary value, one more than the maximum number of tasks in this class that can be queued awaiting initial dispatch. Queuing can occur either because the number of active tasks in the class is already at the MAXACTIVE value or because the maximum for the system has been reached (see the MAXTASKS option in the INQUIRE SYSTEM command). Tasks that arrive while the queue is at its PURGETHRESH limit are purged (abended with a code of AKCC).

The PURGETHRESH value for a class can be between 0-1000000. A value of zero means there is no purge threshold limit; that is, any number of tasks can be queued. A value of one means that no tasks can be queued.

Raising the PURGETHRESH limit allows more transactions to queue and has an effect only when a task is attached that would have been purged if the old value were in effect.

However, if you lower the PURGETHRESH limit beyond the current size of the queue, enough queued tasks are abended to reduce the queue to the new limit. If you raise MAXACTIVE at the same time you lower PURGETHRESH, CICS dispatches as many queued tasks as possible before purging queued tasks, to minimize the number of tasks that get abended. Tasks are abended in priority order, starting with the lowest priority task.

TRANCLASS(*data-value*)

specifies the 8-character name of the transaction class that you are changing. If the class is one of the numbered classes used in earlier releases of CICS, its name is DFHTCL*nm*, where *nm* is the two-digit class number.

Conditions

INVREQ

RESP2 values:

- 2 The MAXACTIVE value is not in the range 0-999.
- 3 The PURGETHRESH value is not in the range 0-1000000.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

TCIDERR

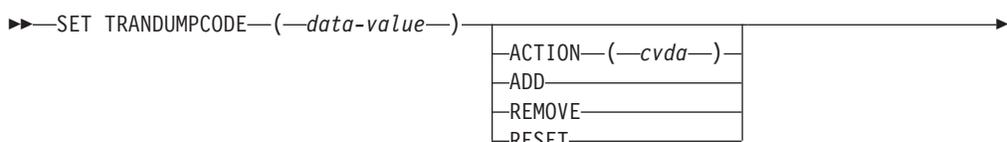
RESP2 values:

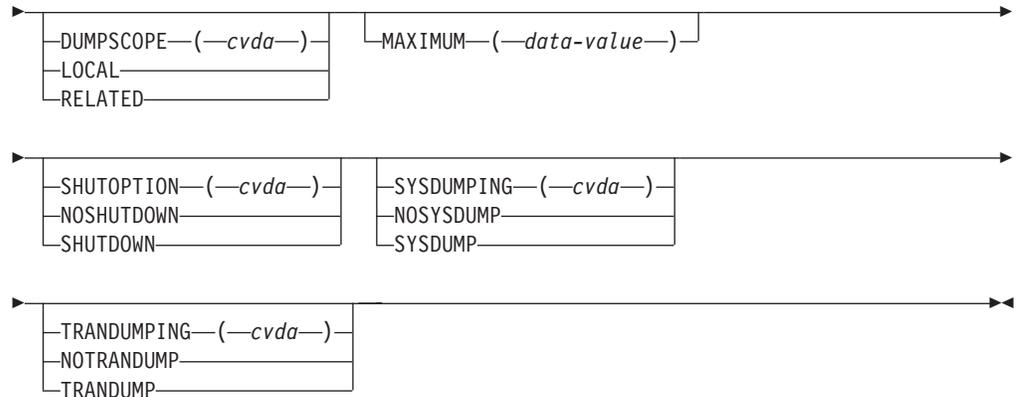
- 1 The transaction class cannot be found.

SET TRANDUMPCODE

Change an entry in the transaction dump table.

SET TRANDUMPCODE





Conditions: DUPREC, INVREQ, IOERR, NOSPACE, NOTAUTH, NOTFND

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The SET TRANDUMPCODE command allows you to change the transaction dump table entry for a particular dump code, to add a new dump code to the table, or to delete one.

The table entry tells CICS the actions to take when a transaction dump request with this code is received. Possible actions include taking a transaction dump, taking a system dump (an MVS SDUMP), initiating requests for SDUMPs of related CICS regions, and shutting down CICS. The table entry also indicates how many times this set of actions is to be taken (the MAXIMUM value); after the maximum is reached, requests are counted but otherwise ignored.

Table updates are recorded in the CICS global catalog and preserved over executions of CICS until an initial or cold start occurs, except in the case of temporary table entries. CICS creates a temporary entry when it receives a dump request with a code for which there is no table entry; these entries, and any changes to them, last only for the current execution of CICS. If you want preserve changes to a temporary entry over restarts, you need to remove the dump code from the table and then add it back.

For information about transaction dumps, see the *CICS Problem Determination Guide*.

Options

ACTION(*cvda*)

specifies the action to be taken for the dump code. CVDA values are:

ADD An entry for this code is to be added to the table.

REMOVE

The entry for this code is to be removed from the table. No other options can be specified on a REMOVE request.

RESET

The current number of dump requests for this dump code is to be set to zero. (See the CURRENT option of the INQUIRE TRANDUMPCODE command.)

DUMPSCOPE(*cvda*)

specifies whether a request for a dump with this dump code should cause CICS to initiate requests for SDUMPs (system dumps) of "related" CICS regions.

A related CICS region is one in the same sysplex, connected by MRO/XCF and doing work on behalf of the task that caused the dump request - specifically, a region that has a task doing work under the same APPC token as this task.

This propagation of SDUMP requests occurs only when the table entry for this code also specifies a SYSDUMPING value of SYSDUMP, and only in a sysplex environment executing under MVS/ESA 5.1 and the MVS workload manager. In other systems, specifying RELATED causes an exception condition.

CVDA values are:

LOCAL

SDUMP requests are not to be sent.

RELATED

SDUMP requests are to be sent.

Note: A setting of DUMPSCOPE(RELATED) results in a single dump being taken for each affected MVS image. This dump contains the output from all the affected CICS regions in the image. For more information, see Automatic dump data capture from related CICS regions, in the *CICS Problem Determination Guide*.

LOCAL is the default for entries you add, if you do not specify a DUMPSCOPE value.

MAXIMUM(*data-value*)

specifies, as a fullword binary value, the maximum number of times CICS should take the set of actions indicated in the dump table entry. After the maximum is reached, CICS counts but otherwise ignores dump requests with this code. The valid range is 0-999. A value of 999 means there is no limit, and is the default used if you omit this option from an ADD request.

SHUTOPTION(*cvda*)

specifies whether the CICS system is to be shut down after a request for a dump with this dump code. CVDA values are:

NOSHUTDOWN

The system is not to be shut down.

SHUTDOWN

The system is to be shut down.

If this option is omitted from an ADD request, NOSHUTDOWN is assumed.

SYSDUMPING(*cvda*)

specifies whether a system dump (an MVS SDUMP) should be taken when a transaction dump request with this code is received. CVDA values are:

NOSYSDDUMP

A system dump is not to be taken.

SYSDUMP

A system dump is to be taken.

Even when SYSDUMP is specified, CICS takes a dump only if the number of requests for this code is less than the MAXIMUM and system dumps are not suppressed globally (see the DUMPING option of the INQUIRE SYSTEM command).

If this option is omitted from an ADD request, NOSYSDUMP is assumed.

TRANDUMPCODE (*data-value*)

specifies the 4-character transaction dump code for which the transaction dump table entry is to be changed. A valid transaction dump code has no leading or imbedded blanks.

TRANDUMPING (*cvda*)

specifies whether a transaction dump should be taken when a transaction dump request with this code is received. CVDA values are:

NOTRANDUMP

A transaction dump is not to be taken.

TRANDUMP

A transaction dump is to be taken.

Even when TRANDUMP is specified, CICS will dump only when the count of requests for this code is no greater than the MAXIMUM.

If this option is omitted from an ADD request, TRANDUMP is assumed.

Conditions

DUPREC

RESP2 values:

- 10 ADD is specified for a dump code already in the transaction dump table.

INVREQ

RESP2 values:

- 2 ACTION has an invalid CVDA value.
- 3 TRANDUMPING has an invalid CVDA value.
- 4 SYSDUMPING has an invalid CVDA value.
- 5 The MAXIMUM value is out of range.
- 6 SHUTOPTION has an invalid CVDA value.
- 7 REMOVE is specified with other options.
- 9 The dump code is invalid.
- 13 DUMPSCOPE has an invalid CVDA value.
- 14 RELATED requires MVS/ESA 5.1.

IOERR

RESP2 values:

- 11 An error occurred updating the CICS catalog. The entry is changed for the current run, but is not recorded for restarts.

NOSPACE

RESP2 values:

- 12 The CICS catalog is full. The entry is changed for the current run, but is not recorded for restarts.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

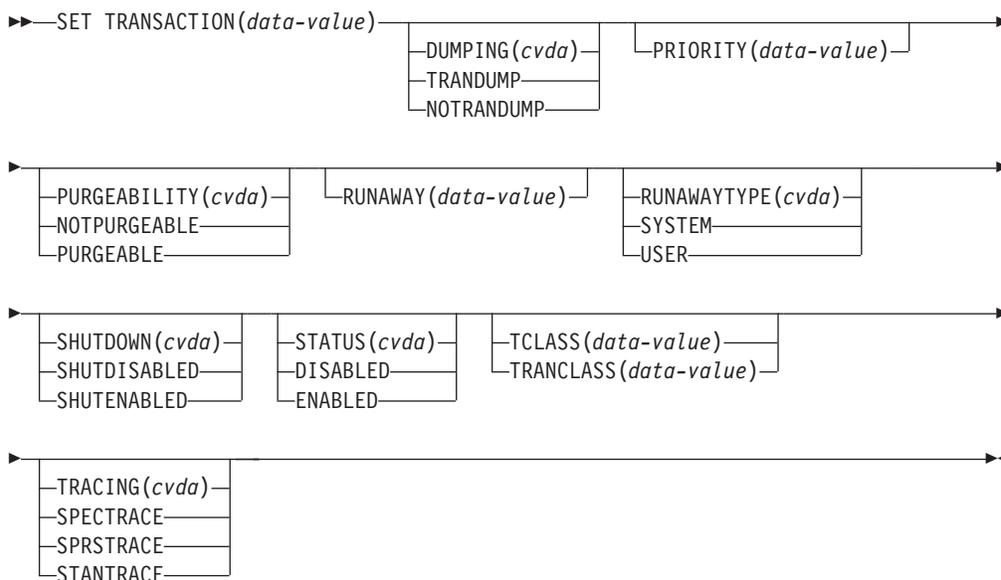
RESP2 values:

- 1 The dump code cannot be found.

SET TRANSACTION

Change a TRANSACTION definition.

SET TRANSACTION



Conditions: INVREQ, NOTAUTH, TRANSIDERR

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The SET TRANSACTION command allows you to change some attributes of a transaction definition.

You can change only the definitions in the local CICS system with this command. If you change a transaction that executes remotely (that is, one that specifies a REMOTESYSTEM value), your changes are made, but they have no effect on the definition in the remote system to which the local definition points, and therefore no effect on tasks that execute the transaction.

Changing a transaction definition affects only future tasks; to change a task already executing the transaction, use the SET TASK command.

Options

DUMPING(*cvda*)

specifies whether CICS should take a transaction dump if a task executing this transaction terminates abnormally. CVDA values are:

NOTRANDUMP

No dump should be taken.

TRANDUMP

A dump should be taken.

This value applies only toabend dumps and has no effect on DUMP TRANSACTION commands.

OTSTIMEOUT(*data-area*)

returns a fullword data-area containing the default period in seconds an OTS transaction, created in an EJB environment executing under this CICS transaction, is allowed to execute before syncpoint.

PRIORITY(*data-value*)

specifies, as a fullword binary value, the priority of this transaction relative to other transactions in the CICS system. The value must be in the range 0–255.

PURGEABILITY(*cvda*)

returns a CVDA value indicating whether CICS is allowed to purge this task (that is, to terminate it abnormally). Purge requests come from SET TASK PURGE commands (or CEMT equivalents), and CICS can generate them internally to reclaim resources to relieve a system stall condition. CVDA values are:

NOTPURGEABLE

The task cannot be purged.

PURGEABLE

The task can be purged.

The PURGEABILITY value is set initially by the SPURGE option in the definition of the transaction this task is executing.

RUNAWAY(*data-value*)

specifies, as a fullword binary value, the “runaway task” time, in milliseconds, for tasks executing this transaction. The value must be 0, or in the range 500–2700000. When checking whether a task is in a runaway condition, CICS rounds the value you specify downwards, to a multiple of 500. If a task keeps control of the processor for more than this interval, CICS assumes it is in a loop and abends it. If the value is zero, CICS does not monitor the task for a runaway condition.

Note: If you specify RUNAWAY, you must set RUNAWAYTYPE to USER in the same SET command, even if RUNAWAYTYPE already has a value of USER.

RUNAWAYTYPE(*cvda*)

specifies where the runaway task time for a task executing this transaction should be obtained. CVDA values are:

SYSTEM

The system default for runaway task time should be used. (An INQUIRE SYSTEM command with the RUNAWAY option tells you what the system value is.)

USER The RUNAWAY value for this transaction should be used. You must specify a value for RUNAWAY when you specify USER.

SHUTDOWN (*cvda*)

specifies whether this transaction can be executed during CICS shutdown by a task created to process unsolicited terminal input. (The transaction also can be executed in this situation if it appears in the transaction list table (XLT) for shutdown.) CVDA values are:

SHUTDISABLED

The transaction cannot be executed.

SHUTENABLED

The transaction can be executed.

STATUS (*cvda*)

specifies whether the transaction is to be available for use. CVDA values are:

DISABLED

The transaction is not available for use.

ENABLED

The transaction is available for use.

Transactions beginning with the letter "C" are CICS-supplied and cannot be disabled.

TCLASS (*data-value*)

specifies, as a fullword binary value, the transaction class to which the transaction is to belong. SET TRANSACTION TCLASS sets the TRANCLASS value in a TRANSACTION definition.

TCLASS is provided only for compatibility with earlier releases of CICS, where transaction classes were numbered rather than named, and you can use it only to assign a name of the form DFHTCL*nn*, where *nn* is the number you specify, in the range 0-10. (It does not change the TCLASS value in the TRANSACTION definition, which CICS maintains for situations in which the same TRANSACTION definition is used for several different releases. See the descriptions of TCLASS and TRANCLASS in the INQUIRE TRANSACTION command for more information.)

TRACING (*cvda*)

specifies the type of tracing to be done for tasks executing this transaction. See the *CICS Problem Determination Guide* for definitions of tracing types. CVDA values are:

SPECTRACE

Tracing is to be special.

SPRSTRACE

Tracing is to be suppressed.

STANTRACE

Tracing is to be standard.

TRANCLASS (*data-value*)

specifies the 8-character name of the transaction class to which this transaction is to belong.

TRANSACTION(*data-value*)

specifies the 4-character name of the transaction definition that you are changing.

Conditions**INVREQ**

RESP2 values:

- 2 PURGEABILITY has an invalid CVDA value.
- 3 STATUS has an invalid CVDA value.
- 4 DISABLED has been specified for a CICS-supplied transaction.
- 5 The TCLASS or TRANCLASS name is not known.
- 7 TRACING has an invalid CVDA value.
- 8 DUMPING has an invalid CVDA value.
- 9 The PRIORITY value is out of range.
- 10 RUNAWAYTYPE has an invalid CVDA value.
- 11 SHUTDOWN has an invalid CVDA value.
- 12 USER has been specified without a RUNAWAY value.
- 13 RUNAWAY has been specified without a RUNAWAYTYPE value of USER.
- 14 The RUNAWAY value is out of range.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

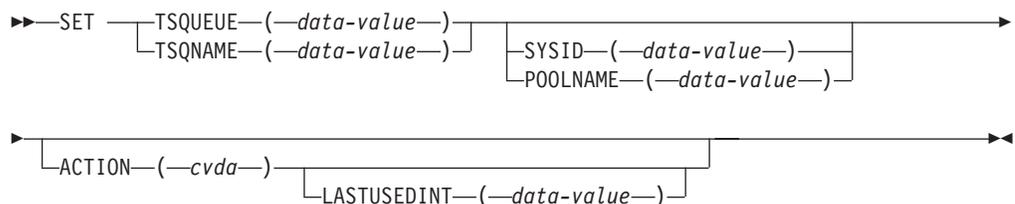
TRANSIDERR

RESP2 values:

- 1 The transaction cannot be found.

SET TSQUEUE / TSQNAME

Delete a TS queue. You can also use the alternative command **SET TSQNAME**. Use either command to delete a queue with a name up to 8 characters long and use SET TSQNAME to delete a queue with a name up to 16 characters long.

SET TSQUEUE

Conditions: INVREQ, NOTAUTH, NOTFND, POOLERR, QIDERR, SYSIDERR

Description

The SET TSQUEUE command enables you to delete a TS queue. The LASTUSEDINT option may be used to ensure that the queue to be deleted has not been referenced since a previous INQUIRE was issued. It may also be used to

delete queues which have not been referenced within a given interval. If a queue is recoverable, a separate task must be attached to perform the deletion.

Options

ACTION(*cvda*)

Specifies the action to be taken on the queue. The CVDA value is:

DELETE

The queue is to be deleted.

LASTUSEDINT

If this option is specified, the queue is deleted only if its last used interval is greater or equal to the value specified. The value is specified in seconds.

POOLNAME(*data-value*)

Specifies an 8-character pool name.

SYSID(*data-value*)

Specifies a 4-character shared SYSID.

TSQNAME(*data-value*)

Specifies the 1 to 16-character identifier of the TS queue.

TSQUEUE(*data-value*)

Specifies the 1 to 8-character identifier of the TS queue.

Conditions

INVREQ

RESP2 values:

- 1 The TSQUEUE was not deleted because LASTUSEDINT was greater than the interval, or because the TSQUEUE is in use.
- 2 The action specified was not *DELETE*
- 3 LASTUSEDINT was specified but had an invalid value; that is, a negative value.
- 4 The file type was not valid for the specified operation.

NOTAUTH

RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

NOTFND

RESP2 values:

- 1 The TSQUEUE cannot be found.

POOLERR

RESP2 values:

- 0 POOLNAME was specified but the pool could not be accessed.

QIDERR

RESP2 values:

- 1 The QUEUE name was invalid; (it was binary zeros).

SYSIDERR

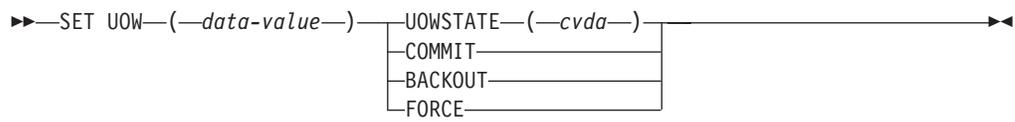
RESP2 values:

- 0 SYSID was specified but there is no corresponding pool, or the pool is unavailable.
- 3 The SYSID does not map to a shared pool.
- 4 Server error.
- 5 I/O error on coupling facility.

SET UOW

Commit, back out, or force a shunted unit of work.

SET UOW



Conditions: INVREQ, NOTAUTH, UOWNOTFOUND

Description

The SET UOW command enables you to commit, back out, or force a unit of work that has been shunted during the indoubt period of the transaction.

Options

UOW(*data-value*)

Specifies the 16-byte identifier of the UOW to be committed, backed out, or forced.

UOWSTATE(*cvda*)

Specifies the action to be attempted for this UOW. CVDA values are as follows:

BACKOUT

Attempt to force syncpoint backout processing, as specified for this UOW.

COMMIT

Attempt to force syncpoint commit processing, as specified for this UOW.

FORCE

Attempt to force the UOW to back out or commit, as specified on the ACTION option of the TRANSACTION resource definition.

Note: All these values are valid only for UOWs that are shunted indoubt. For information about the INDOUBT attributes of TRANSACTION definitions, see the *CICS Resource Definition Guide*.

Conditions

INVREQ

RESP2 values:

- 3 UOWSTATE has an invalid CVDA value.

- 4 CICS is not in a valid state to COMMIT, BACKOUT, or FORCE this UOW.

NOTAUTH

RESP2 values:

- 100 The use of this command is not authorized.

UOWNOTFOUND

RESP2 values:

- 1 The UOW cannot be found.

SET UOWLINK

Delete a link to a unit of work (a UOW-link) that was created by a connection that has since been discarded. UOWLINKs associated with RRS can be deleted when RRS has cold started.

SET UOWLINK

```
▶▶ SET UOWLINK(--data-value--) ACTION(--cvda--) DELETE
```

Conditions: INVREQ, NOTAUTH, UOWNOTFOUND

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

The association between a unit of work and a connection is known as a UOW-link. You can use the INQUIRE UOWLINK command to browse all the UOW-links currently in the system. Some of the UOW-links may have been created by connections that have since been discarded. If so, you may be able to use the SET UOWLINK command to delete them. (For information about when it is safe to delete UOW-links, see the *CICS Intercommunication Guide*).

Options

ACTION(*cvda*)

specifies the action to be taken against the UOW-link. The CVDA value is:

DELETE

Delete the UOW-link. Note that you cannot delete UOW-links where the connection still exists.

UOWLINK(*data-value*)

specifies the 4-character identifier of the unit of work-connection dependency (the UOW-link) to which this command applies.

Conditions

INVREQ

RESP2 values:

- 2 Resynchronization is already in progress, or the UOW-link is already being processed by another instance of the SET UOWLINK command.
- 3 The unit of work is indoubt, and the UOW-link is the coordinator of

the commit or backout session. The unit of work must be forced using the SET UOW command before the UOW-link can be deleted.

- 4 This is not a link created by a connection, or is not a recoverable link.
- 5 The UOW-link (and the associated communication session) is still active.
- 6 ACTION has an invalid CVDA value.
- 7 The UOW-link has a suitable connection definition, and cannot be deleted. You must discard the related connection before you can delete a UOW link.

NOTAUTH

RESP2 values:

- 100 The use of this command is not authorized.

UOWLNOTFOUND

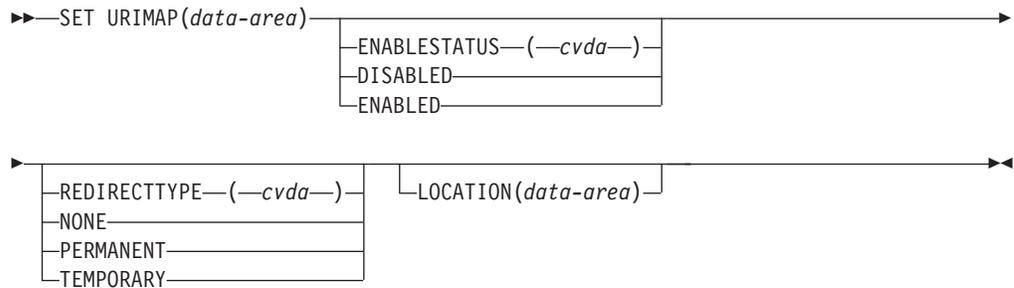
RESP2 values:

- 1 The specified UOW-link cannot be found.

SET URIMAP

Enables or disables a URIMAP definition, and applies or removes redirection for a URIMAP definition.

SET URIMAP



Conditions: INVREQ, NOTAUTH, NOTFND

This command is threadsafe.

Description

The SET URIMAP command allows you to:

- Enable or disable a URIMAP definition.
- Set redirection for matching HTTP requests, and specify a URL to which the requests are redirected. You can use this command to apply redirection to an existing URIMAP definition, for example if the application that would normally respond to the HTTP request is unavailable. You can also use this command to remove redirection from a URIMAP definition.

Options

ENABLESTATUS(cvda)

Sets the URIMAP definition to enabled or disabled status. CVDA values are:

ENABLED

The URIMAP definition can be accessed by applications.

DISABLED

The URIMAP definition cannot be accessed by applications. A URIMAP definition has to be disabled before it can be reinstalled or discarded.

If a URIMAP resource associated with the current HTTP request is disabled, error message DFHWP0763 is issued.

LOCATION (*data-area*)

Specifies a URL of up to 255 characters, to which matching HTTP requests from Web clients can be redirected. This must be a complete URL, including scheme, host, and path components, and appropriate delimiters. CICS does not check that the URL is valid, so you must ensure that the destination exists and that the URL is specified correctly.

The REDIRECTTYPE option is used to specify the type of redirection. If temporary or permanent redirection is specified, the URL in the LOCATION attribute is used for redirection. If NONE is specified, the URL in the LOCATION option is ignored.

REDIRECTTYPE (*cvda*)

Specifies the type of redirection for requests that match this URIMAP definition. The URL for redirection is specified by the LOCATION option. CVDA values are:

NONE

Requests are not redirected. Any URL specified by the LOCATION option is ignored.

TEMPORARY

Requests are redirected on a temporary basis. The HTTP status code used for the response is 302 (Found).

PERMANENT

Requests are redirected permanently. The HTTP status code used for the response is 301 (Moved Permanently).

Conditions**INVREQ**

RESP2 values are:

- 8 No location specified for redirection (LOCATION option).
- 9 Invalid REDIRECTTYPE or ENABLESTATUS value.
- 12 The URIMAP has USAGE(CLIENT) so redirection does not apply, and the LOCATION option cannot be set.

NOTAUTH

RESP2 values are:

- 100 The user associated with the issuing task is not authorized to use this command.

NOTFND

RESP2 values are:

- 3 The URIMAP cannot be found.

SET VOLUME

SET VOLUME is obsolete, and is retained only for compatibility with previous releases. The only runtime support is to return the VOLIDERR condition. If this command is used, the translator translates it, but issues a warning message.

Conditions

VOLIDERR

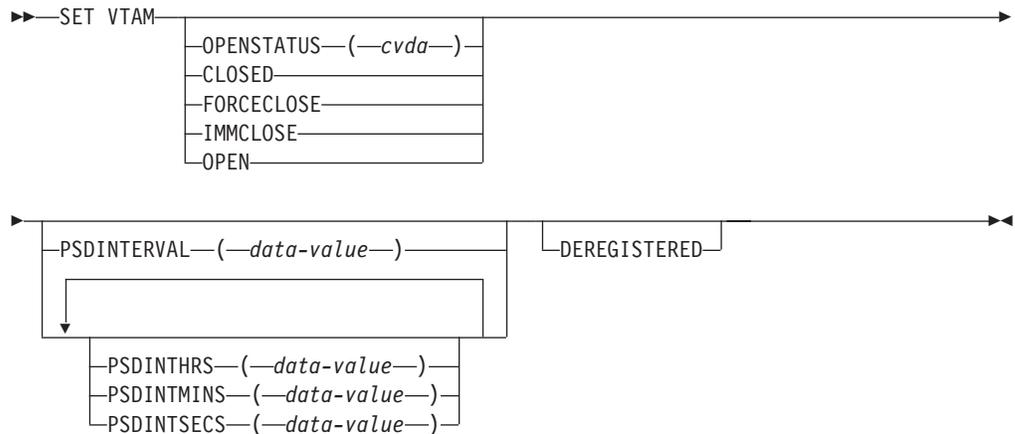
RESP2 values:

- 1 The program has issued a SET VOLUME command. This command is withdrawn.

SET VTAM

Modify the z/OS Communications Server connection for CICS. VTAM is the previous name for z/OS Communications Server.

SET VTAM



Conditions: INVREQ, IOERR, NOTAUTH

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

Description

Use the SET VTAM command to complete the following tasks:

- Establish or stop the CICS connection to the Communications Server.
- Modify the persistent session delay interval value that CICS passes to the Communications Server.
- Deregister CICS from membership of a Communications Server generic resource.

Options

DEREGISTERED

Specifies that CICS is to be removed from the Communications Server generic resource of which it is currently a member. If you deregister a region from membership of a generic resource, you must end any affinities that it owns; see the **PERFORM ENDAFFINITY** command.

Generic resources are described in Workload balancing in a sysplex in the Intercommunication Guide.

OPENSTATUS (*cvda*)

Specifies whether or not CICS is to have a connection to the Communications Server (that is, whether the Communications Server ACB is to be open or closed) and, if CICS must close the ACB to comply, how the shutdown will be performed. CVDA values are as follows:

CLOSED

The connection is to be closed. If it is currently open, CICS is to quiesce all Communications Server activity and then close the Communications Server ACB. Tasks using Communications Server SNA LUs or sessions are allowed to complete before closure, but new tasks requiring the Communications Server are not begun.

FORCECLOSE

The connection is to be closed. If currently open, CICS is to close the Communications Server ACB immediately. Both Communications Server sessions and tasks using the Communications Server end abnormally as a result.

IMMCLOSE

The connection is to be closed. If currently open, CICS is to end abnormally any tasks using the Communications Server immediately, perform an orderly shutdown of all its Communications Server sessions, and then close the Communications Server ACB.

OPEN A connection is to be open. If the Communications Server ACB is closed, CICS is to open it.

If CICS is using the Communications Server multinode persistent sessions, and the Communications Server has been restarted after an abend, opening the Communications Server ACB causes CICS to restore the persistent sessions that the Communications Server has retained. However, CICS does not restore APPC synclevel 2 sessions, which are unbound.

PSDINTERVAL (*data-value*)

Specifies the persistent session delay interval value, which states if, and for how long, the Communications Server is to hold sessions in a recovery-pending state if a failure occurs. The range for the value is 0 – 23:59:59. The value of PSDINTERVAL is not recorded in the global catalog.

If zero is set, sessions are not retained and are stopped at the time of the failure, so persistent sessions support is not exploited.

- If you specify SNPS (the default) or MNPS for the **PSTYPE** system initialization parameter for the CICS region, set a nonzero value for the persistent session delay interval, so that sessions are retained.
- If you specify NOPS (no persistent sessions support) for the **PSTYPE** system initialization parameter, a zero value is required for the persistent session delay interval.

When you specify a persistent session delay interval, CICS sets the **PSDINT** system initialization parameter. CICS passes this value to the Communications Server whenever it opens the ACB. The value is passed immediately if you specify an OPENSTATUS value of OPEN in the same SET SYSTEM command, or if the Communications Server ACB is already open and you do not close it. If the ACB is closed or being closed, or if the open attempt fails, the new value is established on the next successful open.

If the Communications Server is below the level that supports persistent sessions, the Communications Server rejects the request. CICS then sets the PSDINT system option value to zero and returns an INVREQ condition, but goes on to continue any other processing that you requested. The INVREQ occurs when the value is passed to the Communications Server, which might be later than the command that set it, as explained above. Consequently, you can see this condition on a command that does not specify a persistent session delay interval.

The persistent session delay interval can be expressed in several ways:

- A 4-byte packed decimal composite, in the format *Ohmmss+*, using the PSDINTERVAL option.
- With separate hours, minutes, and seconds, using the PSDINTHRS, PSDINTMINS, and PSDINTSECS options. You can use these options singly or in any combination.

When you use PSDINTERVAL or more than one of the separate options, the minutes and seconds portions of the value must not be greater than 59. PSDINTMINS or PSDINTSECS used alone can exceed 59. For example, you can express an interval of 1 hour and 30 minutes in any of the following ways:

- PSDINTERVAL(13000)
- PSDINTHRS(1), PSDINTMINS(30)
- PSDINTMINS(90)
- PSDINTSECS(5400)

PSDINTHRS (*data-value*)

Specifies the hours component of the persistent session delay interval, in fullword binary form. See the PSDINTERVAL option.

PSDINTMINS (*data-value*)

Specifies the minutes component of the persistent session delay interval, in fullword binary form. See the PSDINTERVAL option.

PSDINTSECS (*data-value*)

Specifies the seconds component of the persistent session delay interval, in fullword binary form. See the PSDINTERVAL option.

Conditions

INVREQ

RESP2 values:

- 1 The Communications Server is not present in the system.
- 2 OPENSTATUS has an invalid CVDA value.
- 4 The PSDINTERVAL value is out of range.
- 5 The PSDINTHRS value is out of range.
- 6 The PSDINTMINS value is out of range.
- 7 The PSDINTSECS value is out of range.
- 8 A PSDINTERVAL value > 0 was specified in an XRF-eligible system.
- 9 The Communications Server reported an error while an attempt was being made to set the persistent session delay interval.
- 10 A persistent session delay interval has been specified but either the Communications Server currently in use (or the Communications Server library used when the terminal control table was assembled)

does not support persistent sessions. The interval might have been specified earlier than this command; see the PSDINTERVAL description. If OPEN was also requested, CICS has opened the Communications Server ACB.

- 11 The ACB has opened successfully, but an error occurred in at least one of the sessions that persisted from the previous failure.
- 12 Your OPEN request did not complete because another task subsequently requested a close of the Communications Server connection.
- 13 An error occurred during recovery of sessions, and the Communications Server ACB is closed as a result.
- 14 CICS is performing cleanup processing following a predatory XRF takeover. CICS rejects OPEN requests with this error, without invoking the Communications Server, during this activity. OPEN requests are processed as usual as soon as cleanup is complete.
- 16 Your attempt to deregister CICS from a Communications Server generic resource failed because CICS is not registered as a member of a generic resource group.
- 22 An attempt was made to change PSDINTERVAL, PSDINTHRS, PSDINTMINS, or PSDINTSECS to a nonzero value with the system initialization parameter PSTYPE=NOPS in effect. The request was rejected.

IOERR

RESP2 values:

- n* An error occurred during the opening of the ACB. If CICS was unable to process the request, the RESP2 value is 3. If the Communications Server detected the failure, CICS returns the FDBK2 code in RESP2: you can look up these errors in *z/OS Communications Server: SNA Programming*, under ACB OPEN and CLOSE return codes.

NOTAUTH

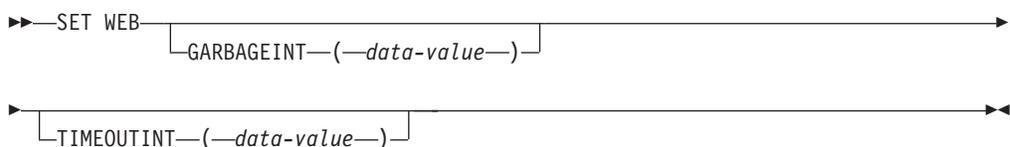
RESP2 values:

- 100 The user associated with the issuing task is not authorized to use this command.

SET WEB

Modify CICS Web support.

SET WEB



Conditions: INVREQ, NOTAUTH

This command is threadsafe.

For additional background information about the SPI commands that handle work requests, see “INQUIRE WORKREQUEST” on page 571.

Options

WORKREQUEST (*data-value*)

Specifies an 8-character token to identify the work request. The token is generated by CICS. Normal usage begins with an **INQUIRE WORKREQUEST BROWSE** command, to identify the token which is of interest. Then the specific token can be used to issue **INQUIRE** and **SET** commands that relate to the task in question.

PURGETYPE (*cvda*)

Specifies that CICS is to purge the task, and indicates conditions for doing so.

Purging a task at the wrong time can result in a loss of data integrity or, in some circumstances, can cause CICS to abend. CICS always defers purging until the task reaches a state where the system itself does not appear to be at risk, but you can specify whether CICS also should wait until data integrity can be ensured.

If CICS accepts a purge request, it returns a **NORMAL** response to **SET WORKREQUEST**. You can tell whether execution has been deferred by inspecting the **RESP2** value. If **RESP2** is zero, the purge has been completed; if **RESP2** is 6, it has been deferred. **CVDA** values are:

FORCEPURGE

The task is to be terminated as soon as it is consistent with system integrity and without regard to data integrity. CICS cannot always determine whether a forced purge is safe; it is possible to abend the system when you specify **FORCEPURGE**.

PURGE

The task is to be terminated as soon as both system and data integrity can be maintained. You cannot purge a task with this **CVDA** value if the definition of the **TRANSACTION** it is executing specifies **SPURGE=NO**.

WORKTYPE (*cvda*)

Specifies the type of work being performed:

IIOP Specifies that the work is being executed for an **IIOP** request.

SOAP Specifies that the work is being executed for a Web service request.

If you do not set a value for this option, the type of work defaults to **IIOP**.

Conditions

INVREQ

RESP2 values:

- 2 The attempted purge or force purge action is not valid.
- 5 The attempt to purge or force purge a local task failed.
- 6 The attempt to purge or force purge a local task is deferred.
- 7 **PURGE** is not permitted because **SPURGE=NO** has been set to protect the task.
- 8 The task is protected by CICS and is not eligible for modification with this command.

- 9 An attempt to purge or force purge a request receiver failed. Such actions are not allowed for request receivers.

NOTAUTH

RESP2 values:

- 100 The user of the issuing task is not authorized to use this command.

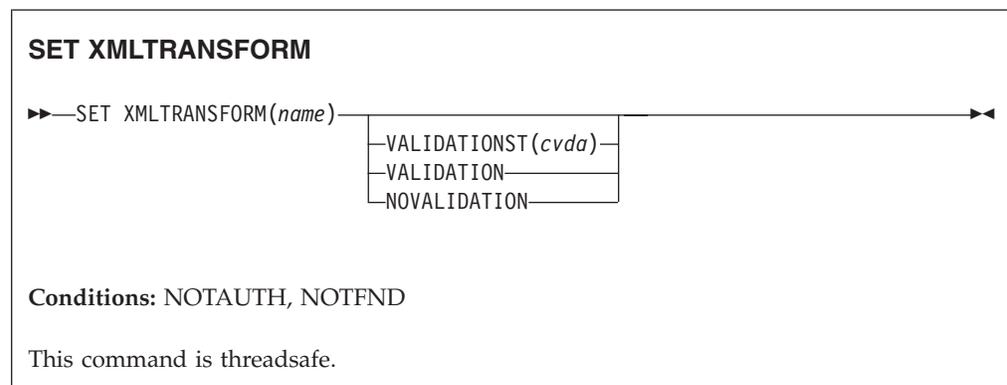
NOTFND

RESP2 values:

- 3 The specified WORKREQUEST is not present in this system.

SET XMLTRANSFORM

Use the **SET XMLTRANSFORM** command to enable or disable the validation of an installed XMLTRANSFORM resource.



Description

CICS dynamically creates an XMLTRANSFORM resource when you install a BUNDLE or ATOMSERVICE resource. The XMLTRANSFORM resource defines the location of the XML binding and schema in z/OS UNIX. Use the **SET XMLTRANSFORM** command to enable or disable validation. If enabled, CICS checks that the XML is valid against the schema.

Options

VALIDATIONST(*cvda*)

Indicates whether full validation is enabled for the XMLTRANSFORM resource. CVDA values are as follows:

VALIDATION

Full validation is enabled.

NOVALIDATION

Full validation is disabled.

Because validating an XML message against its schema incurs considerable processing overhead, typically you will specify VALIDATIONST(NOVALIDATION). If VALIDATIONST(NOVALIDATION) is specified, checking is performed to ensure that the message contains well-formed XML, but with no guarantee that the XML is valid.

Full validation ensures that the XML in the message is valid with respect to the XML schema; you might want to specify `VALIDATIONST(VALIDATION)` when you are developing an application.

XMLTRANSFORM(*name*)

Specifies the 1-32 character name of the XMLTRANSFORM resource.

Conditions

NOTAUTH

RESP2 values are:

- 100 The user associated with the issuing task is not authorized to use this command.
- 101 The user associated with the issuing task is not authorized to access this particular resource in the way required by this command.

NOTFND

RESP2 values are:

- 3 The XMLTRANSFORM cannot be found.

Appendix A. CICS-value data areas used by all commands

The CICS-value data-area (CVDA) values and their numeric equivalents for all of the EXEC CICS commands.

CVDA's are described in CICS-value data areas (CVDA's) in CICS System Programming Reference.

CVDA's and numeric values in alphabetic sequence

CVDA	Value
ABEND	900
ACQFAIL	515
ACQUIRED	69
ACQUIRING	71
ACTIVE	181
ACTIVITY	1002
ADD	291
ADDABLE	41
ADDFAIL	519
ADDRESS	895
ADVANCE	265
AINIT	282
ALARM	501
ALL	856
ALLCONN	169
ALLOCATD	81
ALLQUERY	431
ALLVALUES	860
ALTERABLE	52
ALTERNATE	197
ALTPRTCOPY	446
ANALYZER	1126
AND	1005
ANY	158
APLKYBD	391
APLTEXT	393
APPC	124
APPCPARALLEL	374
APPCSINGLE	373
APPEND	1036
APPLICATION	559

CVDA	Value
APPLID	303
APPLNAME	338
ASACTL	224
ASCII7	616
ASCII8	617
ASSEMBLER	150
ASSERTED	1104
ASYNCHRONOUS	570
ATI	75
ATOM	1119
ATOMSERVICE	1179
ATTACH	576
ATTENTION	524
AUDALARM	395
AUTOACTIVE	630
AUTOARCH	262
AUTOAUTH	1095
AUTOCONN	170
AUTOINACTIVE	631
AUTOINIT	284
AUTOINSTALL	1140
AUTOPAGE	80
AUTOREGISTER	1094
AUTOSTART	618
AUTOTERMID	1120
AUXILIARY	247
AUXPAUSE	313
AUXSTART	312
AUXSTOP	314
AVAILABLE	95
BACKOUT	192
BACKTRANS	397
BACKUPNONBWO	800
BASE	10
BASEAPI	1052
BASESPACE	664
BASE64	50
BASICAUTH	1092
BDAM	2
BEGINSESSION	510
BELOW	159

CVDA	Value
BGAM	63
BINARY	1038
BINIT	283
BIT	1020
BLK	47
BLOCKED	16
BOTH	1100
BRIDGE	935
BROWSABLE	39
BSAM	61
BUNDLE	1180
BUSY	612
C	149
CACHE	791
CANCEL	526
CANCELLED	624
CANCELLING	1025
CAPTURESPEC	1195
CATEGORY	474
CBE	582
CBER	581
CCE	583
CD	491
CEDF	370
CERTIFICAUTH	1093
CERTUSER	1061
CFTABLE	833
CFE	584
CGROUP	902
CHANNEL	1047
CHAR	1019
CHUNKEND	737
CHUNKNO	735
CHUNKYES	736
CICS	660
CICSAPI	1052
CICSDATAKEY	379
CICSECURITY	195
CICSEXECKEY	381
CICSTABLE	101
CKOPEN	1055

CVDA	Value
CLASSCACHE	1075
CLEAR	640
CLICONVERT	743
CLIENT	1122
CLIENTAUTH	1032
CLIENTCERT	1087
CLOSE	741
CLOSED	19
CLOSEFAILED	349
CLOSELEAVE	261
CLOSEREQ	22
CLOSING	21
CMDPROT	673
CMDSECEXT	207
CMDSECNO	205
CMDSECYES	207
COBOL	151
COBOLII	375
COLD	788
COLDACQ	72
COLDQUERY	433
COLDSTART	266
COLLECTION	473
COLOR	399
COMMAREA	1048
COMMIT	208
COMMITFAIL	792
COMPAT	107
COMPLETE	1026
COMPOSITE	1003
COMPRESS	1130
CONFFREE	82
CONFRECV	83
CONFSEND	84
CONNECT	903
CONNECTED	690
CONNECTING	904
CONNECTION	755
CONSISTENT	723
CONSOLE	66
CONTAINER	869

I

CVDA	Value
CONTENTION	836
CONTEXT	574
CONTROLSHUT	623
CONVERSE	600
CONVIDLE	518
COORDINATOR	770
COPID	908
COPY	401
CORBA	1101
CORBASERVER	1141
CPLUSPLUS	148
CREATE	67
CREATESPI	844
CRITICAL	11
CSDAPI	840
CSDBATCH	841
CSIGN	905
CTERM	906
CTLGALL	632
CTLGMODIFY	633
CTLGNONE	634
CTRLABLE	56
CTX	907
CURRENT	260
CURRENTPGM	870
CUSERID	909
CUSTOM	579
DAEOPT	684
DATA	508
DATASET	756
DATASETFULL	793
DATASTREAM	543
DB2CONN	1142
DB2ENTRY	1143
DB2TRAN	1184
DEADLOCK	794
DEBUG	1082
DEC	46
DEFAULT	198
DEFAULTUSER	1070
DEFINITION	990

I

CVDA	Value
DEFRESP1	497
DEFRESP1OR2	528
DEFRESP2	498
DEFRESP3	499
DELAY	637
DELETABLE	43
DELETE	292
DELETEDFAIL	520
DELEXITERROR	795
DEREGERROR	679
DEREGISTERED	678
DEST	235
DISABLED	24
DISABLEDHOST	1125
DISABLING	25
DISCARDFAIL	513
DISCARDING	1118
DISCONNING	910
DISCREQ	444
DISK1	252
DISK2	253
DISK2PAUSE	254
DISPATCHABLE	228
DISPATCHER	1144
DJAR	1189
DOCDELETE	295
DOCTEMPLATE	1145
DOESNOTEQUAL	861
DOESNOTEXIST	872
DOESNOTSTART	862
DORMANT	1024
DPLSUBSET	383
DRAINING	619
DREPAPI	843
DREPBATCH	845
DS3270	615
DUALCASE	403
DUMMY	781
DUPERROR	106
DUPNOREPLACE	105
DUPREPLACE	104

|
|
|

CVDA	Value
DYNAMIC	178
EB	490
EBCDIC	1039
ECI	1098
ECIIP	936
EJB	1102
EMERGENCY	268
EMPTY	210
EMPTYREQ	31
ENABLED	23
ENABLING	51
ENDAFFINITY	790
ENQMODEL	1188
ENQUEUE	1146
EPADAPTER	1196
EQUAL	911
EQUALS	863
ESDS	5
EVENT	334
EVENTBINDING	1191
EVENTFAIL	1012
EVENTPROCESS	1192
EVENTUAL	3
EXCEPT	332
EXCEPTRESP	523
EXCI	650
EXCTL	48
EXECENQ	751
EXECENQADDR	752
EXECINSTALL	846
EXISTS	871
EXIT	912
EXITTRACE	362
EXPECT	747
EXPIRED	1017
EXPORT	991
EXTENDED	1128
EXTENDEDDES	405
EXTRA	221
EXTSECURITY	194
FAILED	782

CVDA	Value
FAILEDBKOUT	357
FAILINGBKOUT	358
FAULTTOEPR	853
FCLOSE	273
FEED	472
FILE	238
FINALQ	183
FINPUT	270
FIRE	1001
FIRSTINIT	625
FIRSTQ	182
FIXED	12
FLUSH	783
FMH	502
FMHPARM	385
FOPEN	272
FORCE	342
FORCECANCEL	655
FORCECLOSE	351
FORCED	1013
FORCEPURGE	237
FORCEUOW	342
FORCLOSING	353
FORMATEDF	606
FORMATTED	542
FORMFEED	407
FOUTPUT	271
FREE	85
FREEING	94
FROMEPR	850
FULL	212
FULLAPI	384
FULLAUTO	1073
FWDRECOVABLE	354
GENERIC	651
GET	748
GMT	604
GOHIGHERTHAN	873
GOINGOUT	172
GOLOWERTHAN	874
GREATERTHAN	864

CVDA	Value
GROUP	913
GRPLIST	842
GTFSTART	317
GTFSTOP	318
HEAD	750
HEURBACKOUT	758
HEURCOMMIT	757
HEX	45
HFORM	409
HFSFILE	1040
HIGH	914
HILIGHT	413
HOLD	163
HOME	1103
HOSTNAME	302
HTTP	1096
HTTPNO	1034
HTTPS	1029
HTTPYES	1033
IDENTIFY	1074
IDNTY	328
IGNORE	1
IGNORED	1105
IGNORERR	816
IIOP	1097
IMMCLOSE	350
IMMCLOSING	352
IMMEDIATE	2
IMMQUIESCED	706
IMPORT	992
INACTIVE	378
INBOUND	547
INCOMPLETE	1014
INDEXRECFULL	796
INDIRECT	122
INDOUBT	620
INFLIGHT	621
INITCOMPLETE	628
INITIAL	789
INITING	1112
INOUT	532

CVDA	Value
INPUT	226
INSERVICE	73
INSTALLED	550
INSTALLFAIL	512
INTERNAL	1058
INTRA	222
INTSTART	310
INTSTOP	311
INVALID	359
IOERROR	797
IPCONN	1176
IPECI	936
IPIC	1089
IPV4	300
IPV6	301
IRC	121
ISNOTGREATER	865
ISNOTLESS	866
ISOLATE	658
JAVA	147
JOURNALMODEL	1185
JOURNALNAME	1147
JOURNALNUM	1148
JVM	1080
JVMPOOL	1149
JVMPROFILE	1150
JVMPROGRAM	1151
JVMSERVER	1193
KATAKANA	415
KEEP	786
KEYED	8
KILL	232
KSDS	6
LCKSTRUCFULL	832
LEAVE	811
LENV	377
LESSTHAN	867
LE370	377
LIBRARY	1177
LIC	493
LIGHTPEN	417

CVDA	Value
LINK	577
LOAD	834
LOCAL	605
LOCKING	837
LOG	54
LOGICAL	216
LOGTERM	269
LOSE	544
LOW	915
LPA	165
LSRPOOL	1152
LUP	541
LUSTAT	525
LU61	125
MAIN	248
MAP	155
MAPSET	155
METADATA	857
MCHCTL	241
MDT	506
MNPS	999
MOD	813
MODEANY	189
MODEL	370
MODE24	190
MODE31	191
MONITOR	1153
MORE	492
MQCONN	1175
MRO	938
MSRCONTROL	419
MTOM	460
MTOMNOXOP	469
MVS	780
MVSTCB	1154
NEGATIVE	530
NEW	28
NEWCOPY	167
NEWSSESSION	485
NOALARM	500
NOALTPRTCOPY	447

CVDA	Value
NOANALYZER	1127
NOAPLKYBD	392
NOAPLTEXT	394
NOAPPEND	1037
NOAPPLNAME	339
NOATI	76
NOAUDALARM	396
NOAUTHENTIC	1091
NOAUTO	1071
NOAUTOARCH	263
NOBACKOUT	193
NOBACKTRANS	398
NOCEDF	371
NOCLASSCACHE	1076
NOCLEAR	641
NOCLICONVERT	744
NOCLOSE	742
NOCMDPROT	674
NOCOLOR	400
NOCOMPAT	108
NOCOMPRESS	1131
NOCONNECT	916
NOCONV	556
NOCONVERSE	601
NOCOPY	402
NOCREATE	68
NOCTL	223
NODAEOPT	685
NODE	1155
NODEBUG	1083
NODISCREQ	445
NODOCDELETE	296
NODUALCASE	404
NOEMPTYREQ	32
NOEVENT	335
NOEXCEPT	333
NOEXCTL	49
NOEXIT	917
NOEXITTRACE	363
NOEXTENDED DS	406
NOFMH	503

CVDA	Value
NOFMHPARM	386
NOFORCE	1054
NOFORMATEDF	607
NOFORMFEED	408
NOHFORM	410
NOHILIGHT	414
NOHOLD	164
NOIDNTY	329
NOINCONVERT	745
NOISOLATE	657
NOJVM	1081
NOKATAKANA	416
NOKEEP	787
NOLIGHTPEN	418
NOLOAD	835
NOLOG	55
NOLOSTLOCKS	710
NOMDT	507
NOMSGJRNL	531
NOMSRCONTROL	420
NOMTOM	461
NOMTOMNOXOP	470
NONAUTOCONN	171
NONCICS	661
NONCRITICAL	1090
NONE	496
NONLE370	1084
NOOBFORMAT	422
NOOBOPERID	388
NOOUTCONVERT	746
NOOUTLINE	424
NOPARTITIONS	426
NOPERF	331
NOPHASEOUT	1110
NOPRESETSEC	243
NOPRINTADAPT	428
NOPROGSYMBOL	430
NOPRTCOPY	449
NOPS	997
NOQUERY	432
NORECOVDATA	700

CVDA	Value
NOREENTPROT	681
NORELEASE	918
NORELREQ	443
NORESRC	337
NORESVC	934
NORETAINED	711
NOREUSE	294
NORMAL	1016
NORMALBKOUT	356
NORMALRESP	522
NORMI	775
NOROLLBACK	919
NOSECURITY	196
NOENDDMTOM	464
NOSHUTDOWN	289
NOSOSI	435
NOSPI	694
NOSRVCONVERT	739
NOSSL	1031
NOSTSN	487
NOSWITCH	285
NOSYNCPOINT	603
NOSYSCONNECT	654
NOSYSDUMP	185
NOSYSLOG	784
NOTADDABLE	42
NOTALLOWED	327
NOTALTERABLE	53
NOTAPPLIC	1
NOTASKSTART	608
NOTBRWBLE	40
NOTBUSY	613
NOTCDEB	495
NOTCONNECTED	691
NOTCTRLABLE	57
NOTDEFINED	659
NOTDELBLE	44
NOTDYNAMIC	1021
NOTEMPTY	211
NOTEXTENDED	1129
NOTERMINAL	214

CVDA	Value
NOTEXTKYBD	437
NOTEXTPRINT	439
NOTFIRED	1000
NOTFWDRCVBLE	361
NOTINBOUND	546
NOTINIT	376
NOTINSTALLED	551
NOTKEYED	9
NOTLPA	166
NOTPEND	127
NOTPURGEABLE	161
NOTRANDUMP	187
NONTRANS	572
NOTREADABLE	36
NOTREADY	259
NOTRECOVABLE	30
NOTREQUIRED	667
NOTRLS	721
NOTROUTABLE	1022
NOTSOS	669
NOTSUPPORTED	15
NOTSUSPENDED	1027
NOTTABLE	100
NOTTI	78
NOTUPDBLE	38
NOTWAIT	920
NOUCTRAN	451
NOUSER	1011
NOVALIDATION	441
NOVFORM	412
NOWAIT	341
NOWRITE	275
NOXOPDIRECT	466
NOXOPSUPPORT	468
NOZCPTRACE	365
NRS	774
N906	931
N906D	921
OBFORMAT	421
OBOPERID	387
OBTAINING	96

CVDA	Value
OFF	200
OK	274
OLD	26
OLDCOPY	162
OLDSESSION	486
ON	201
OPEN	18
OPENAPI	1053
OPENERROR	798
OPENING	20
OPENINPUT	256
OPENOUTPUT	257
OPID	933
OPTIONAL	326
OPTIONS	759
OR	1006
OTS	1107
OUTLINE	423
OUTPUT	227
OUTSERVICE	74
OWNER	753
PAGEABLE	79
PARTITIONS	425
PARTITIONSET	156
PARTNER	1182
PATH	11
PDSMEMBER	1035
PENDBEGIN	558
PENDDATA	560
PENDFREE	86
PENDING	126
PENDINIT	1113
PENDPASS	565
PENDRECEIVE	87
PENDRELEASE	562
PENDRESOLVE	1114
PENDSTART	561
PENDSTSN	557
PENDUNSOL	564
PERF	330
PERMANENT	484

CVDA	Value
PHASEIN	168
PHASEOUT	1111
PHYSICAL	215
PIPELINE	1124
PLI	152
PL1	152
POOL	922
POSITIVE	529
POST	636
POSTCOMMAND	697
PRECOMMAND	696
PRESETSEC	242
PRIMARY	110
PRINCIPAL	199
PRINTADAPT	427
PRIVATE	174
PROCESS	1010
PROCESSTYPE	1190
PROFILE	157
PROGAUTO	1072
PROGRAM	154
PROGRAMDEF	1178
PROGRAMINIT	698
PROGSYMBOL	429
PROTECTED	504
PROVIDER	1041
PRTCOPY	448
PUBLISH	1085
PURGE	236
PURGEABLE	160
PUT	749
QR	1057
QUASIRENT	1050
QUEUE	814
QUIESCED	707
QUIESCING	708
READABLE	35
READBACK	209
READONLY	275
READY	258
REBUILD	923

CVDA	Value
RECEIVE	88
RECEIVER	1046
RECONNECT	924
RECOVDATA	701
RECOVERABLE	29
RECOVERED	277
RECOVERLOCKS	712
RECOVERY	1156
REENTPROT	680
REFPARMS	858
REGERROR	677
REGION	575
REGISTERED	670
REJECT	815
RELATED	675
RELEASE	563
RELEASED	70
RELEASING	549
RELOAD	1078
RELOADING	1079
RELREQ	442
REMLOSTLOCKS	713
REMOTE	4
REMOVE	276
REMSSESSION	740
REMTABLE	103
REPEATABLE	724
REPLYTOEPR	852
REQCONTEXT	854
REQUEST	249
REQUESTER	1042
REQUESTMODEL	1157
REQUIRED	666
REREREAD	812
RESET	290
RESETLOCKS	714
RESOLVING	1115
RESRCE	336
RESSECEXT	204
RESSECINT	203
RESSECNO	202

CVDA	Value
RESSECYES	204
RESPCONTEXT	855
RESPECTED	1109
RESYNC	702
RETAINED	715
RETRACT	1086
RETRY	716
REUSE	293
REVERTED	264
REWIND	811
RFC1123	646
RFC3339	647
RLS	720
RLSACTIVE	730
RLSGONE	799
RLSINACTIVE	731
RLSSERVER	761
RMI	771
ROLLBACK	89
ROUTABLE	1023
ROUTE	638
RPG	153
RRCOMMITFAIL	830
RRDS	7
RRINDOUBT	831
RRMS	1063
RRSUR	939
RTR	527
RU	494
RUNNING	229
RZINSTOR	940
SAMESENDMTOM	462
SCAN	1088
SCHEDULER	941
SCS	614
SECONDINIT	626
SECUSER	1060
SEND	90
SENDER	1045
SENDMTOM	463
SERVER	1123

CVDA	Value
SERVICE	475
SESSION	372
SESSIONFAIL	517
SESSIONLOST	516
SESSIONS	1187
SETFAIL	514
SHARE	27
SHARED	173
SHUNTED	762
SHUTDISABLED	645
SHUTDOWN	288
SHUTENABLED	644
SIGN	925
SIGNEDOFF	245
SIGNEDON	244
SINGLEOFF	324
SINGLEON	323
SKIP	810
SMF	255
SNA	60
SNPS	998
SOAP	1044
SOCKET	942
SOS	668
SOSABOVE	683
SOSBELOW	682
SOSI	434
SPECIAL	98
SPECIFIC	652
SPECTRACE	177
SPI	693
SPRSTRACE	175
SQLCODE	926
SRVCONVERT	738
SSL	1030
STANDBY	629
STANTRACE	176
START	635
STARTED	609
STARTTERM	943
STARTING	772

CVDA	Value
STARTSWITH	868
STARTUP	180
STATIC	179
STATS	1158
STOPPED	610
STORAGE	1159
STREAMNAME	1160
STSN	509
STSNSET	488
STSNTEST	489
SUBORDINATE	773
SUBPOOL	1161
SUBSPACE	663
SUMMUNLIKE	566
SUPPORTED	1106
SURROGATE	371
SUSPENDED	231
SWITCH	188
SWITCHALL	287
SWITCHING	225
SWITCHNEXT	286
SYNCFREE	91
SYNCHRONOUS	571
SYNCPOINT	602
SYNCRECV	92
SYNCSEND	93
SYSCONNECT	653
SYSDUMP	184
SYSDUMPCODE	1162
SYSLOG	785
SYSPLEX	662
SYSTEM	643
SYSTEMOFF	320
SYSTEMON	319
TABLE	847
TABLEMGR	1163
TAKEOVER	111
TAPE1	250
TAPE2	251
TASK	233
TASKSUBPOOL	1164

CVDA	Value
TASKSTART	611
TCAM	64
TCAMSNA	65
TCEXITALL	366
TCEXITALLOFF	369
TCEXITNONE	368
TCEXITSYSTEM	367
TCLASS	239
TCPIP	802
TCPIPSERVICE	1166
TDQ	767
TDQUEUE	767
TEMPLATENAME	1168
TEMPORARY	483
TERM	234
TERMINAL	213
TERMINATE	1077
TEXTKYBD	436
TEXTPRINT	438
THIRDINIT	627
THREADSAFE	1051
TIME	240
TIMEOUT	511
TIMER	1004
TOEPR	851
TPOOL	932
TPS55M2	552
TPS55M3	553
TPS55M4	554
TPS55M5	555
TRACE	760
TRANCLASS	1169
TRANDATA	944
TRANDUMP	186
TRANDUMPCODE	1170
TRANIDONLY	452
TRANS	573
TRANSACTION	1171
TRANSTART	580
TSMODEL	1183
TSQ	768

I

CVDA	Value
TSQUEUE	768
TTI	77
TWAIT	927
TX	929
TXID	928
TYPETERM	1186
T3278M2	533
T3278M3	534
T3278M4	535
T3278M5	536
T3279M2	537
T3279M3	538
T3279M4	539
T3279M5	540
UCTRAN	450
UKOPEN	1056
UNATTEMPTED	820
UNAVAILABLE	672
UNBLOCKED	17
UNCOMMITTED	722
UNCONNECTED	703
UNDEFINED	14
UNDETERMINED	355
UNENABLED	33
UNENABLING	34
UNEXPECTED	1015
UNEXPIRED	1018
UNKNOWN	774
UNPROTECTED	505
UNQUIESCED	709
UNREGISTERED	671
UNRESOLVED	1116
UNSOLDATA	521
UNUSABLE	1117
UOW	246
UPDATABLE	37
UPDATING	1049
URIMAP	1173
URMTERMID	1121
USER	642
USERDATAKEY	380

CVDA	Value
USEREXECKEY	382
USERID	930
USEROFF	322
USERON	321
USERTABLE	102
VALID	360
VALIDATION	440
VARIABLE	13
VERIFY	1099
VFORM	411
VRRDS	732
VSAM	3
VTAM	60
WAIT	340
WAITCOMMIT	763
WAITER	754
WAITFORGET	622
WAITING	765
WAITRMI	766
WARMSTART	267
WBE	585
WEB	945
WEBSERVICE	1174
WIN	545
WMQ	578
XCF	665
XM	123
XMLTRANSFORM	1194
XMRUNTRAN	946
XNOTDONE	144
XOK	143
XOPDIRECT	465
XOPSUPPORT	467
XPLINK	1068
ZCPTRACE	364

Note: VTAM is now z/OS Communications Server.

CVDAs and numeric values in numeric sequence

Value	CVDA
1	IGNORE
1	NOTAPPLIC
2	BDAM
2	IMMEDIATE
3	EVENTUAL
3	VSAM
4	REMOTE
5	ESDS
6	KSDS
7	RRDS
8	KEYED
9	NOTKEYED
10	BASE
11	CRITICAL
11	PATH
12	FIXED
13	VARIABLE
14	UNDEFINED
15	NOTSUPPORTED
16	BLOCKED
17	UNBLOCKED
18	OPEN
19	CLOSED
20	OPENING
21	CLOSING
22	CLOSEREQUEST
23	ENABLED
24	DISABLED
25	DISABLING
26	OLD
27	SHARE
28	NEW
29	RECOVERABLE
30	NOTRECOVABLE
31	EMPTYREQ
32	NOEMPTYREQ
33	UNENABLED
34	UNENABLING
35	READABLE

Value	CVDA
36	NOTREADABLE
37	UPDATABLE
38	NOTUPDATABLE
39	BROWSABLE
40	NOTBROWSABLE
41	ADDABLE
42	NOTADDABLE
43	DELETABLE
44	NOTDELETABLE
45	HEX
46	DEC
47	BLK
48	EXCTL
49	NOEXCTL
50	BASE64
51	ENABLING
52	ALTERABLE
53	NOTALTERABLE
54	LOG
55	NOLOG
56	CTRLABLE
57	NOTCTRLABLE
60	SNA
60	VTAM
61	BSAM
63	BGAM
64	TCAM
65	TCAMSNA
66	CONSOLE
67	CREATE
68	NOCREATE
69	ACQUIRED
70	RELEASED
71	ACQUIRING
72	COLDACQ
73	INSERVICE
74	OUTSERVICE
75	ATI
76	NOATI
77	TTI
78	NOTTI

Value	CVDA
79	PAGEABLE
80	AUTOPAGEABLE
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
94	FREEING
95	AVAILABLE
96	OBTAINING
100	NOTTABLE
101	CICSTABLE
102	USERTABLE
103	REMTABLE
110	PRIMARY
111	TAKEOVER
121	IRC
122	INDIRECT
123	XM
124	APPC
125	LU61
126	PENDING
127	NOTPENDING
143	XOK
144	XNOTDONE
147	JAVA
148	CPLUSPLUS
149	C
150	ASSEMBLER
151	COBOL
152	PLI
152	PL1
153	RPG

Value	CVDA
154	PROGRAM
155	MAP
155	MAPSET
156	PARTITIONSET
157	PROFILE
158	ANY
159	BELOW
160	PURGEABLE
161	NOTPURGEABLE
162	OLDCOPY
163	HOLD
164	NOHOLD
165	LPA
166	NOTLPA
167	NEWCOPY
168	PHASEIN
169	ALLCONN
170	AUTOCONN
171	NONAUTOCONN
172	GOINGOUT
173	SHARED
174	PRIVATE
175	SPRSTRACE
176	STANTRACE
177	SPECTRACE
178	DYNAMIC
179	STATIC
180	STARTUP
181	ACTIVE
182	FIRSTQUIESCE
183	FINALQUIESCE
184	SYSDUMP
185	NOSYSDUMP
186	TRANDUMP
187	NOTRANDUMP
188	SWITCH
189	MODEANY
190	MODE24
191	MODE31
192	BACKOUT
193	NOBACKOUT

Value	CVDA
194	EXTSECURITY
195	CICSECURITY
196	NOSECURITY
197	ALTERNATE
198	DEFAULT
199	PRINCIPAL
200	OFF
201	ON
202	RESSECNO
203	RESSECINT
204	RESSECEXT
204	RESSECYES
205	CMDSECNO
207	CMDSECEXT
207	CMDSECYES
208	COMMIT
209	READBACK
210	EMPTY
211	NOTEMPTY
212	FULL
213	TERMINAL
214	NOTERMINAL
215	PHYSICAL
216	LOGICAL
221	EXTRA
222	INTRA
223	NOCTL
224	ASACTL
225	SWITCHING
226	INPUT
227	OUTPUT
228	DISPATCHABLE
229	RUNNING
231	SUSPENDED
232	KILL
233	TASK
234	TERM
235	DEST
236	PURGE
237	FORCEPURGE
238	FILE

Value	CVDA
239	TCLASS
240	TIME
241	MCHCTL
242	PRESETSEC
243	NOPRESETSEC
244	SIGNEDON
245	SIGNEDOFF
246	UOW
247	AUXILIARY
248	MAIN
249	REQUEST
250	TAPE1
251	TAPE2
252	DISK1
253	DISK2
254	DISK2PAUSE
255	SMF
256	OPENINPUT
257	OPENOUTPUT
258	READY
259	NOTREADY
260	CURRENT
261	CLOSELEAVE
262	AUTOARCH
263	NOAUTOARCH
264	REVERTED
265	ADVANCE
266	COLDSTART
267	WARMSTART
268	EMERGENCY
269	LOGTERM
270	FINPUT
271	FOUTPUT
272	FOPEN
273	FCLOSE
274	OK
275	NOWRITE
275	READONLY
276	REMOVE
277	RECOVERED
282	AINIT

Value	CVDA
283	BINIT
284	AUTOINIT
285	NOSWITCH
286	SWITCHNEXT
287	SWITCHALL
288	SHUTDOWN
289	NOSHUTDOWN
290	RESET
291	ADD
292	DELETE
293	REUSE
294	NOREUSE
300	IPV4
301	IPV6
302	HOSTNAME
303	APPLID
310	INTSTART
311	INTSTOP
312	AUXSTART
313	AUXPAUSE
314	AUXSTOP
317	GTFSTART
318	GTFSTOP
319	SYSTEMON
320	SYSTEMOFF
321	USERON
322	USEROFF
323	SINGLEON
324	SINGLEOFF
326	OPTIONAL
327	NOTALLOWED
328	IDNTY
329	NOIDNTY
330	PERF
331	NOPERF
332	EXCEPT
333	NOEXCEPT
334	EVENT
335	NOEVENT
336	RESRCE
337	NORESRCE

Value	CVDA
338	APPLNAME
339	NOAPPLNAME
340	WAIT
341	NOWAIT
342	FORCE
342	FORCEUOW
349	CLOSEFAILED
350	IMMCLOSE
351	FORCECLOSE
352	IMMCLOSING
353	FORCECLOSING
354	FWDRECOVABLE
355	UNDETERMINED
356	NORMALBKOUT
357	FAILEDDBKOUT
358	FAILINGBKOUT
359	INVALID
360	VALID
361	NOTFWDRCVBLE
362	EXITTRACE
363	NOEXITTRACE
364	ZCPTRACE
365	NOZCPTRACE
366	TCEXITALL
367	TCEXITSYSTEM
368	TCEXITNONE
369	TCEXITALLOFF
370	CEDF
370	MODEL
371	NOCEDF
371	SURROGATE
372	SESSION
373	APPC SINGLE
374	APPC PARALLEL
375	COBOLII
376	NOTINIT
377	LENV
377	LE370
378	INACTIVE
379	CICS DATAKEY
380	USER DATAKEY

Value	CVDA
381	CICSEXECKEY
382	USEREXECKEY
383	DPLSUBSET
384	FULLAPI
385	FMHPARM
386	NOFMHPARM
387	OBOPERID
388	NOOBOPERID
391	APLKYBD
392	NOAPLKYBD
393	APLTEXT
394	NOAPLTEXT
395	AUDALARM
396	NOAUDALARM
397	BACKTRANS
398	NOBACKTRANS
399	COLOR
400	NOCOLOR
401	COPY
402	NOCOPY
403	DUALCASE
404	NODUALCASE
405	EXTENDEDDES
406	NOEXTENDEDDES
407	FORMFEED
408	NOFORMFEED
409	HFORM
410	NOHFORM
411	VFORM
412	NOVFORM
413	HILIGHT
414	NOHILIGHT
415	KATAKANA
416	NOKATAKANA
417	LIGHTPEN
418	NOLIGHTPEN
419	MSRCONTROL
420	NOMSRCONTROL
421	OBFORMAT
422	NOOBFORMAT
423	OUTLINE

Value	CVDA
424	NOOUTLINE
425	PARTITIONS
426	NOPARTITIONS
427	PRINTADAPT
428	NOPRINTADAPT
429	PROGSYMBOL
430	NOPROGSYMBOL
431	ALLQUERY
432	NOQUERY
433	COLDQUERY
434	SOSI
435	NOSOSI
436	TEXTKYBD
437	NOTEXTKYBD
438	TEXTPRINT
439	NOTEXTPRINT
440	VALIDATION
441	NOVALIDATION
442	RELREQ
443	NORELREQ
444	DISCREQ
445	NODISCREQ
446	ALTPRTCOPY
447	NOALTPRTCOPY
448	PRTCOPY
449	NOPRTCOPY
450	UCTRAN
451	NOUCTRAN
452	TRANIDONLY
460	MTOM
461	NOMTOM
462	SAMESENDMTOM
463	SENDMTOM
464	NOSENDMTOM
465	XOPDIRECT
466	NOXOPDIRECT
467	XOPSUPPORT
468	NOXOPSUPPORT
469	MTOMNOXOP
470	NOMTOMNOXOP
472	FEED

Value	CVDA
473	COLLECTION
474	CATEGORY
475	SERVICE
483	TEMPORARY
484	PERMANENT
485	NEWSSESSION
486	OLDSESSION
487	NOSTSN
488	STSNSET
489	STSNTEST
490	EB
491	CD
492	MORE
493	LIC
494	RU
495	NOTCDEB
496	NONE
497	DEFRESP1
498	DEFRESP2
499	DEFRESP3
500	NOALARM
501	ALARM
502	FMH
503	NOFMH
504	PROTECTED
505	UNPROTECTED
506	MDT
507	NOMDT
508	DATA
509	STSN
510	BEGINSESSION
511	TIMEOUT
512	INSTALLFAIL
513	DISCARDFAIL
514	SETFAIL
515	ACQFAIL
516	SESSIONLOST
517	SESSIONFAIL
518	CONVIDLE
519	ADDFAIL
520	DELETEDFAIL

Value	CVDA
521	UNSOLDATA
522	NORMALRESP
523	EXCEPTRESP
524	ATTENTION
525	LUSTAT
526	CANCEL
527	RTR
528	DEFRESP1OR2
529	POSITIVE
530	NEGATIVE
531	NOMSGJRNL
532	INOUT
533	T3278M2
534	T3278M3
535	T3278M4
536	T3278M5
537	T3279M2
538	T3279M3
539	T3279M4
540	T3279M5
541	LUP
542	FORMATTED
543	DATASTREAM
544	LOSE
545	WIN
546	NOTINBOUND
547	INBOUND
549	RELEASING
550	INSTALLED
551	NOTINSTALLED
552	TPS55M2
553	TPS55M3
554	TPS55M4
555	TPS55M5
556	NOCONV
557	PENDSTSN
558	PENDBEGIN
559	APPLICATION
560	PENDDATA
561	PENDSTART
562	PENDRELEASE

Value	CVDA
563	RELEASE
564	PENDUNSOL
565	PENDPASS
566	SUMMUNLIKE
570	ASYNCHRONOUS
571	SYNCHRONOUS
572	NONTRANS
573	TRANS
574	CONTEXT
575	REGION
576	ATTACH
577	LINK
578	WMQ
579	CUSTOM
580	TRANSTART
581	CBER
582	CBE
583	CCE
584	CFE
585	WBE
600	CONVERSE
601	NOCONVERSE
602	SYNCPOINT
603	NOSYNCPOINT
604	GMT
605	LOCAL
606	FORMATEDF
607	NOFORMATEDF
608	NOTASKSTART
609	STARTED
610	STOPPED
611	TASKSTART
612	BUSY
613	NOTBUSY
614	SCS
615	DS3270
616	ASCII7
617	ASCII8
618	AUTOSTART
619	DRAINING
620	INDOUBT

Value	CVDA
621	INFLIGHT
622	WAITFORGET
623	CONTROLSHUT
624	CANCELLED
625	FIRSTINIT
626	SECONDINIT
627	THIRDINIT
628	INITCOMPLETE
629	STANDBY
630	AUTOACTIVE
631	AUTOINACTIVE
632	CTLGALL
633	CTLGMODIFY
634	CTLGNONE
635	START
636	POST
637	DELAY
638	ROUTE
640	CLEAR
641	NOCLEAR
642	USER
643	SYSTEM
644	SHUTENABLED
645	SHUTDISABLED
646	RFC1123
647	RFC3339
650	EXCI
651	GENERIC
652	SPECIFIC
653	SYSCONNECT
654	NOSYSCONNECT
655	FORCECANCEL
657	NOISOLATE
658	ISOLATE
659	NOTDEFINED
660	CICS
661	NONCICS
662	SYSPLEX
663	SUBSPACE
664	BASESPACE
665	XCF

Value	CVDA
666	REQUIRED
667	NOTREQUIRED
668	SOS
669	NOTSOS
670	REGISTERED
671	UNREGISTERED
672	UNAVAILABLE
673	CMDPROT
674	NOCMDPROT
675	RELATED
677	REGERROR
678	DEREGISTERED
679	DEREGERROR
680	REENTPROT
681	NOREENTPROT
682	SOSBELOW
683	SOSABOVE
684	DAE
685	NODAE
690	CONNECTED
691	NOTCONNECTED
693	SPI
694	NOSPI
696	PRECOMMAND
697	POSTCOMMAND
698	PROGRAMINIT
700	NORECOVDATA
701	RECOVDATA
702	RESYNC
703	UNCONNECTED
706	IMMQUIESCED
707	QUIESCED
708	QUIESCING
709	UNQUIESCED
710	NOLOSTLOCKS
711	NORETAINED
712	RECOVERLOCKS
713	REMLOSTLOCKS
714	RESETLOCKS
715	RETAINED
716	RETRY

Value	CVDA
720	RLS
721	NOTRLS
722	UNCOMMITTED
723	CONSISTENT
724	REPEATABLE
730	RLSACTIVE
731	RLSINACTIVE
732	VRRDS
735	CHUNKNO
736	CHUNKYES
737	CHUNKEND
738	SRVCONVERT
739	NOSRVCONVERT
740	REMSSESSION
741	CLOSE
742	NOCLOSE
743	CLICONVERT
744	NOCLICONVERT
745	NOINCONVERT
746	NOOUTCONVERT
747	EXPECT
748	GET
749	PUT
750	HEAD
751	EXECENQ
752	EXECENQADDR
753	OWNER
754	WAITER
755	CONNECTION
756	DATASET
757	HEURCOMMIT
758	HEURBACKOUT
759	OPTIONS
760	TRACE
761	RLSSERVER
762	SHUNTED
763	WAITCOMMIT
765	WAITING
766	WAITRMI
767	TDQ (TDQUEUE)
768	TSQ (TSQUEUE)

Value	CVDA
770	COORDINATOR
771	RMI
772	STARTING
773	SUBORDINATE
774	NRS
774	UNKNOWN
775	NORMI
780	MVS
781	DUMMY
782	FAILED
783	FLUSH
784	NOSYSLOG
785	SYSLOG
786	KEEP
788	COLD
789	INITIAL
790	ENDAFFINITY
791	CACHE
792	COMMITFAIL
793	DATASETFULL
794	DEADLOCK
795	DELEXITERROR
796	INDEXRECFULL
797	IOERROR
798	OPENERROR
799	RLSGONE
800	BACKUPNONBWO
802	TCPIP
810	SKIP
811	LEAVE
811	REWIND
812	REREAD
813	MOD
814	QUEUE
815	REJECT
816	IGNORERR
820	UNATTEMPTED
830	RRCOMMITFAIL
831	RRINDOUBT
832	LCKSTRUCFULL
833	CFTABLE

Value	CVDA
907	CTX
908	COPID
909	CUSERID
910	DISCONNING
911	EQUAL
912	EXIT
913	GROUP
914	HIGH
915	LOW
916	NOCONNECT
917	NOEXIT
918	NORELEASE
919	NOROLLBACK
920	NOTWAIT
921	N906D
922	POOL
923	REBUILD
924	RECONNECT
925	SIGN
926	SQLCODE
927	TWAIT
928	TXID
929	TX
930	USERID
931	N906
932	TPOOL
933	OPID
934	NORES SYNC
935	BRIDGE
936	ECIIP
938	MRO
939	RRSUR
940	RZINSTOR
941	SCHEDULER
942	SOCKET
943	STARTTERM
944	TRANDATA
945	WEB
946	XMRUNTRAN
990	DEFINITION
991	EXPORT

Value	CVDA
992	IMPORT
997	NOPS
998	SNPS
999	MNPS
1000	NOTFIRED
1001	FIRED
1002	ACTIVITY
1003	COMPOSITE
1004	TIMER
1005	AND
1006	OR
1010	PROCESS
1011	NOUSER
1012	EVENTFAIL
1013	FORCED
1014	INCOMPLETE
1015	UNEXPECTED
1016	NORMAL
1017	EXPIRED
1018	UNEXPIRED
1019	CHAR
1020	BIT
1021	NOTDYNAMIC
1022	NOTROUTABLE
1023	ROUTABLE
1024	DORMANT
1025	CANCELLING
1026	COMPLETE
1027	NOTSUSPENDED
1029	HTTPS
1030	SSL
1031	NOSSL
1032	CLIENTAUTH
1033	HTTPYES
1034	HTTPNO
1035	PDSMEMBER
1036	APPEND
1037	NOAPPEND
1038	BINARY
1039	EBCDIC
1040	HFSFILE

Value	CVDA
1041	PROVIDER
1042	REQUESTER
1044	SOAP
1045	SENDER
1046	RECEIVER
1047	CHANNEL
1048	COMMAREA
1049	UPDATING
1050	QUASIRENT
1051	THREADSAFE
1052	BASEAPI
1052	CICSAPI
1053	OPENAPI
1054	NOFORCE
1055	CKOPEN
1056	UKOPEN
1057	QR
1058	INTERNAL
1060	SECUSER
1061	CERTUSER
1063	RRMS
1068	XPLINK
1070	DEFAULTUSER
1071	NOAUTO
1072	PROGAUTO
1073	FULLAUTO
1074	IDENTIFY
1075	CLASSCACHE
1076	NOCLASSCACHE
1077	TERMINATE
1078	RELOAD
1079	RELOADING
1080	JVM
1081	NOJVM
1082	DEBUG
1083	NODEBUG
1084	NONLE370
1085	PUBLISH
1086	RETRACT
1087	CLIENTCERT
1088	SCAN

Value	CVDA
1089	IPIC
1090	NONCRITICAL
1091	NOAUTHENTIC
1092	BASICAUTH
1093	CERTIFICAUTH
1094	AUTOREGISTER
1095	AUTOAUTH
1096	HTTP
1097	IIOF
1098	ECI
1099	VERIFY
1100	BOTH
1101	CORBA
1102	EJB
1103	HOME
1104	ASSERTED
1105	IGNORED
1106	SUPPORTED
1107	OTS
1109	RESPECTED
1110	NOPHASEOUT
1111	PHASEOUT
1112	INITING
1113	PENDINIT
1114	PENDRESOLVE
1115	RESOLVING
1116	UNRESOLVED
1117	UNUSABLE
1118	DISCARDING
1119	ATOM
1120	AUTOTERMID
1121	URMTERMID
1122	CLIENT
1123	SERVER
1124	PIPELINE
1125	DISABLEDHOST
1126	ANALYZER
1127	NOANALYZER
1128	EXTENDED
1129	NOTEXTENDED
1130	COMPRESS

Value	CVDA
1131	NOCOMPRESS
1140	AUTOINSTALL
1141	CORBASERVER
1142	DB2CONN
1143	DB2ENTRY
1144	DISPATCHER
1145	DOCTEMPLATE
1146	ENQUEUE
1147	JOURNALNAME
1148	JOURNALNUM
1149	JVMPOOL
1150	JVMPROFILE
1151	JVMPROGRAM
1152	LSRPOOL
1153	MONITOR
1154	MVSTCB
1155	NODE
1156	RECOVERY
1157	REQUESTMODEL
1158	STATS
1159	STORAGE
1160	STREAMNAME
1161	SUBPOOL
1162	SYSDUMPCODE
1163	TABLEMGR
1164	TASKSUBPOOL
1166	TCPIPSERVICE
1168	TEMPLATENAME
1169	TRANCLASS
1170	TRANDUMPCODE
1171	TRANSACTION
1173	URIMAP
1174	WEBSERVICE
1175	MQCONN
1176	IPCONN
1177	LIBRARY
1178	PROGRAMDEF
1180	BUNDLE
1186	TYPETERM
1196	EPADAPTER

I

Note: VTAM is now z/OS Communications Server.

CVDA values for the DEVICE option

CVDA sequence	
BATCHLU	191
BIPROG	160
BISYNCH	128
CDRDLPRT	24
CONTNLU	189
HARDCOPY	32
INTACTLU	190
ISCMCONV	209
LUCMODGRP	210
LUCSESS	211
LUTYPE4	193
LUTYPE6	192
MAGTAPE	20
RESSYS	208
SDLC	176
SEQDISK	18
SYSTEM3	161
SYSTEM7	2
SYS370	164
SYS7BSCA	166
TCONSOLE	8
TELETYPE	34
TTCAM	80
TWX3335	33
T1050	36
T1053	74
T2260L	65
T2260R	72
T2265	76
T2740	40
T2741BCD	43
T2741COR	42
T2770	130
T2780	132
T2980	134
T3275R	146
T3277L	153

CVDA sequence	
T3277R	145
T3284L	155
T3284R	147
T3286L	156
T3286R	148
T3600BI	138
T3601	177
T3614	178
T3650ATT	186
T3650PIPE	184
T3650USER	187
T3653HOST	185
T3735	136
T3740	137
T3780	133
T3790	180
T3790SCSP	182
T3790UP	181
T7770	1
VIDEOTERM	64

Numeric sequence	
1	T7770
2	SYSTEM7
8	TCONSOLE
18	SEQDISK
20	MAGTAPE
24	CDRDLPRT
32	HARDCOPY
33	TWX3335
34	TELETYPE
36	T1050
40	T2740
42	T2741COR
43	T2741BCD
64	VIDEOTERM
65	T2260L
72	T2260R
74	T1053
76	T2265
80	TTCAM

Numeric sequence	
128	BISYNCH
130	T2770
132	T2780
133	T3780
134	T2980
136	T3735
137	T3740
138	T3600BI
145	T3277R
146	T3275R
147	T3284R
148	T3286R
153	T3277L
155	T3284L
156	T3286L
160	BIPROG
161	SYSTEM3
164	SYS370
166	SYS7BSCA
176	SDLC
177	T3601
178	T3614
180	T3790
181	T3790UP
182	T3790SCSP
184	T3650PIPE
185	T3653HOST
186	T3650ATT
187	T3650USER
189	CONTNLU
190	INTACTLU
191	BATCHLU
192	LUTYPE6
193	LUTYPE4
208	RESSYS
209	ISCMCONV
210	LUCMODGRP
211	LUCSESS

Appendix B. EXEC interface block (EIB) response and function codes

This appendix lists the response codes and the function codes of EXEC CICS commands.

Response codes of EXEC CICS commands

After the execution of an EXEC CICS command, fields EIBRESP and EIBRCODE are set to indicate whether the command executed successfully, or whether a CICS condition was raised. Use the response codes to help in problem determination.

Each possible value of EIBRESP relates directly to a specific condition, no matter which command caused the condition to be raised. This is not true for EIBRCODE values: both the value and the byte of EIBRCODE in which it is set depend on which command was issued.

The following sections list the conditions that are applicable to EXEC CICS commands, their corresponding RESP values (decimal), the associated EIBRCODE values (hexadecimal), and the transaction abend codes (if any).

EXEC CICS CSD, DISCARD, INQUIRE, PERFORM, and SET commands

The first word of EIBRCODE for these commands is always set equal to the hexadecimal equivalent of the RESP value; the remaining bytes are set to X'00'.

Condition (Byte 3)	RESP Value code code	EIBRCODE	Abend
CSDERR	119	77	AEZS
DSNNOTFOUND	93	5D	AEX1
DUPREC	14	0E	AEIN
DUPRES	120	78	AEZT
END	83	53	AEXK
FILENOTFOUND	12	0C	AEIL
ILLOGIC	21	15	AEIU
INCOMPLETE	126	7E	AEZZ
INVREQ	16	10	AEIP
IOERR	17	11	AEIQ
JIDERR	43	2B	AEYG
LENGERR	22	16	AEIV
LOCKED	100	64	AEX8
MODELIDERR	95	5F	AEX3
NOSPACE	18	12	AEIR
NOSTG	42	2A	—
NOTAUTH	70	46	AEY7
NOTFND	13	0D	AEIM
PARTNERIDERR	97	61	AEX5
PGMIDERR	27	1B	AEI0
PROFILEIDERR	98	62	AEX6
QIDERR	44	2C	AEYH
SYSBUSY	59	3B	—
SYSIDERR	53	35	AEYQ

Condition (Byte 3)	RESP Value code code	EIBRCODE	Abend
TASKIDERR	91	5B	AEXX
TCIDERR	92	5C	AEX0
TERMIDERR	11	0B	AEIK
TRANSIDERR	28	1C	AEI1
UOWNOTFOUND	102	66	-
USERIDERR	69	45	AEYX
VOLIDERR	71	47	AEXV

EXEC CICS DISABLE, ENABLE, and EXTRACT EXIT commands

Conditions that can be raised by the DISABLE, ENABLE, and EXTRACT EXIT commands are INVEXITREQ and NOTAUTH. There are no conditions associated with the RESYNC command.

Condition	RESP Value	EIBRCODE	Abend
INVEXITREQ	63	80	AEY0
NOTAUTH	70	46	AEY7

Function codes of EXEC CICS commands

The function code (field EIBFN) is a hexadecimal value that identifies the command most recently issued by a task.

The format of the EIBFN field is as follows:

```

ASM      CL2
COBOL   PIC X(2)
PL/I    CHAR (2)
C       CHAR variable name(2);

```

The function codes of the commands described in this book are listed below in command sequence, then in function code sequence.

Command	Code
ACQUIRE TERMINAL	86 02
COLLECT STATISTICS	70 08
CREATE ATOMSERVICE	30 42
CREATE BUNDLE	30 40
CREATE CONNECTION	30 0E
CREATE CORBASERVER	30 32
CREATE DB2CONN	30 20
CREATE DB2ENTRY	30 22
CREATE DB2TRAN	30 24
CREATE DJAR	30 34
CREATE DOCTEMPLATE	30 2E
CREATE ENQMODEL	30 2A
CREATE FILE	30 14
CREATE IPCONN	30 3C
CREATE JOURNALMODEL	30 1E
CREATE JVMSERVER	30 46
CREATE LSRPOOL	30 16
CREATE LIBRARY	30 3E

Command	Code
CREATE MAPSET	30 04
CREATE MQCONN	30 44
CREATE PARTITIONSET	30 06
CREATE PARTNER	30 18
CREATE PIPELINE	30 38
CREATE PROCESSTYPE	30 26
CREATE PROFILE	30 0A
CREATE PROGRAM	30 02
CREATE REQUESTMODEL	30 2C
CREATE SESSIONS	30 12
CREATE TCPIPSERVICE	30 30
CREATE TDQUEUE	30 1C
CREATE TERMINAL	30 10
CREATE TRANCLASS	30 1A
CREATE TRANSACTION	30 08
CREATE TSMODEL	30 28
CREATE TYPETERM	30 0C
CREATE URIMAP	30 36
CREATE WEBSERVICE	30 3A
CSD ADD	A2 02
CSD ALTER	A2 04
CSD APPEND	A2 06
CSD COPY	A2 08
CSD DEFINE	A2 0A
CSD DELETE	A2 0C
CSD DISCONNECT	A2 32
CSD ENDBRGROUP	A2 2C
CSD ENDBRLIST	A2 2E
CSD ENDBRRSRCE	A2 30
CSD GETNEXTGROUP	A2 20
CSD GETNEXTLIST	A2 22
CSD GETNEXTRSRCE	A2 24
CSD INQUIREGROUP	A2 1A
CSD INQUIRELIST	A2 1C
CSD INQUIRERSRCE	A2 1E
CSD INSTALL	A2 0E
CSD LOCK	A2 10
CSD REMOVE	A2 12
CSD RENAME	A2 14
CSD STARTBRGROUP	A2 26
CSD STARTBRLIST	A2 28
CSD STARTBRRSRCE	A2 2A
CSD UNLOCK	A2 16
CSD USERDEFINE	A2 18
DISABLE PROGRAM	22 04
DISCARD ATOMSERVICE	CC 10
DISCARD AUTINSTMODEL	42 10
DISCARD BUNDLE	C8 10
DISCARD CONNECTION	58 10
DISCARD CORBASERVER	B2 10
DISCARD DB2CONN	94 10
DISCARD DB2ENTRY	94 30
DISCARD DB2TRAN	94 50
DISCARD DOCTEMPLATE	9E 10

Command	Code
DISCARD ENQMODEL	90 90
DISCARD FILE	4C 10
DISCARD DJAR	B2 30
DISCARD IPCONN	C2 10
DISCARD JOURNALMODEL	92 10
DISCARD JOURNALNAME	60 10
DISCARD JVMSERVER	B0 10
DISCARD LIBRARY	C6 10
DISCARD MQCONN	CE 10
DISCARD PARTNER	44 10
DISCARD PIPELINE	BC 10
DISCARD PROCESSTYPE	96 10
DISCARD PROFILE	46 10
DISCARD PROGRAM	4E 10
DISCARD REQUESTMODEL	9A 10
DISCARD TCPIPSERVICE	9C 10
DISCARD TDQUEUE	5C 10
DISCARD TERMINAL	52 10
DISCARD TRANCLASS	5E 18
DISCARD TRANSACTION	50 10
DISCARD TSMODEL	80 30
DISCARD URIMAP	BE 10
DISCARD WEBSERVICE	BC 30
ENABLE PROGRAM	22 02
EXTRACT EXIT	22 06
EXTRACT STATISTICS	70 26
INQUIRE ASSOCIATION	C4 02
INQUIRE ATOMSERVICE	CC 02
INQUIRE AUTINSTMODEL	42 02
INQUIRE AUTOINSTALL	68 12
INQUIRE BUNDLE	C8 02
INQUIRE BUNDLEPART	C8 12
INQUIRE BEAN	B2 42
INQUIRE BRFACILITY	B4 02
INQUIRE CAPTURESPEC	CA 22
INQUIRE CFDTPPOOL	98 02
INQUIRE CLASSCACHE	B0 22
INQUIRE CONNECTION	58 02
INQUIRE CORBASERVER	B2 02
INQUIRE DB2CONN	94 02
INQUIRE DB2ENTRY	94 22
INQUIRE DB2TRAN	94 42
INQUIRE DELETSHIPED	68 22
INQUIRE DISPATCHER	B6 02
INQUIRE DJAR	B2 22
INQUIRE DOCTEMPLATE	9E 02
INQUIRE DSNAME	7A 02
INQUIRE DUMPDS	66 02
INQUIRE ENQ	90 22
INQUIRE ENQMODEL	90 82
INQUIRE EVENTBINDING	CA 02
INQUIRE EVENTPROCESS	CA 12
INQUIRE EXCI	7C 02
INQUIRE EXITPROGRAM	88 02

Command	Code
INQUIRE FILE	4C 02
INQUIRE HOST	BE 12
INQUIRE IPCONN	C2 02
INQUIRE IPFACILITY	C2 12
INQUIRE IRC	6E 02
INQUIRE JOURNALMODEL	92 02
INQUIRE JOURNALNAME	60 12
INQUIRE JOURNALNUM	60 02
INQUIRE JVM	B0 32
INQUIRE JVMPOOL	B0 02
INQUIRE JVMPROFILE	B0 12
INQUIRE JVMSERVER	B0 42
INQUIRE LIBRARY	C6 02
INQUIRE MODENAME	5A 02
INQUIRE MONITOR	70 12
INQUIRE MQCONN	CE 02
INQUIRE MQINI	CE 12
INQUIRE MVSTCB	B6 12
INQUIRE NETNAME	52 16
INQUIRE NETNAME	52 06
INQUIRE PARTNER	44 02
INQUIRE PIPELINE	BC 02
INQUIRE PROCESSTYPE	96 02
INQUIRE PROFILE	46 02
INQUIRE PROGRAM	4E 02
INQUIRE REQID	8A 02
INQUIRE REQUESTMODEL	9A 02
INQUIRE RRMS	3A 02
INQUIRE STATISTICS	70 02
INQUIRE STORAGE	5E 08
INQUIRE STREAMNAME	92 12
INQUIRE SUBPOOL	5E 42
INQUIRE SYSDUMPCODE	66 22
INQUIRE SYSTEM	54 02
INQUIRE TASK	5E 02
INQUIRE TCLASS	5E 12
INQUIRE TCPIP	9C 12
INQUIRE TCPIPSERVICE	9C 02
INQUIRE TDQUEUE	5C 02
INQUIRE TEMPSTORAGE	80 32
INQUIRE TERMINAL	52 02
INQUIRE TERMINAL	52 12
INQUIRE TRACEDEST	78 02
INQUIRE TRACEFLAG	78 12
INQUIRE TRACETYPE	78 22
INQUIRE TRANCLASS	5E 1A
INQUIRE TRANDUMPCODE	66 12
INQUIRE TRANSACTION	50 02
INQUIRE TSMODEL	80 22
INQUIRE TSPool	80 1A
INQUIRE TSQNAME	80 12
INQUIRE TSQUEUE	80 02
INQUIRE UOW	90 02
INQUIRE UOWDSNFAIL	90 62

Command	Code
INQUIRE UOWENQ	90 22
INQUIRE UOWLINK	90 42
INQUIRE URIMAP	BE 02
INQUIRE VOLUME	62 02
INQUIRE VTAM	68 02
INQUIRE WEB	9C 22
INQUIRE WEBSERVICE	BC 22
INQUIRE WORKREQUEST	B8 02
INQUIRE XMLTRANSFORM	D0 02
PERFORM CLASSCACHE	B0 26
PERFORM CORBASERVER	B2 06
PERFORM DELETSHIPED	68 26
PERFORM DJAR	B2 24
PERFORM DUMP	7E 04
PERFORM ENDAFFINITY	58 06
PERFORM PIPELINE	BC 06
PERFORM RESETTIME	72 02
PERFORM SECURITY	64 02
PERFORM SHUTDOWN	76 02
PERFORM STATISTICS	70 06
RESYNC ENTRYNAME	16 04
SET ATOMSERVICE	CC 04
SET AUTOINSTALL	68 14
SET BUNDLE	C8 04
SET BRFACILITY	B4 04
SET CLASSCACHE	B0 24
SET CONNECTION	58 04
SET CORBASERVER	B2 04
SET DB2CONN	94 04
SET DB2ENTRY	94 24
SET DB2TRAN	94 44
SET DELETSHIPED	68 24
SET DISPATCHER	B6 04
SET DSNNAME	7A 04
SET DUMPDS	66 04
SET ENQMODEL	90 84
SET EVENTBINDING	CA 04
SET EVENTPROCESS	CA 14
SET FILE	4C 04
SET HOST	BE 14
SET IPCONN	C2 04
SET IRC	6E 04
SET JOURNALNAME	60 14
SET JOURNALNUM	60 04
SET JVMPOOL	B0 04
SET JVMSERVER	B0 44
SET LIBRARY	C6 04
SET MODENAME	5A 04
SET MONITOR	70 14
SET MQCONN	CE 04
SET NETNAME	52 08
SET PIPELINE	BC 04
SET PROCESSTYPE	96 04
SET PROGRAM	4E 04

Command	Code
SET STATISTICS	70 04
SET SYSDUMPCODE	66 24
SET SYSTEM	54 04
SET TASK	5E 04
SET TCLASS	5E 14
SET TCPIP	9C 14
SET TCPIPSERVICE	9C 04
SET TDQUEUE	5C 04
SET TEMPSTORAGE	80 34
SET TERMINAL	52 04
SET TERMINAL	52 14
SET TRACEDEST	78 04
SET TRACEFLAG	78 14
SET TRACETYPE	78 24
SET TRANCLASS	5E 1C
SET TRANDUMPCODE	66 14
SET TRANSACTION	50 04
SET TSQNAME	80 14
SET TSQUEUE	80 04
SET UOW	90 04
SET UOWLINK	90 44
SET URIMAP	BE 04
SET VOLUME	62 04
SET VTAM	68 04
SET WEB	9C 24
SET WEBSERVICE	BC 24
SET WORKREQUEST	B8 04
SET XMLTRANSFORM	D0 04

Note: VTAM is now z/OS Communications Server.

Code	Command
16 04	RESYNC ENTRYNAME
22 02	ENABLE PROGRAM
22 04	DISABLE PROGRAM
22 06	EXTRACT EXIT
30 0A	CREATE PROFILE
30 0C	CREATE TYPETERM
30 0E	CREATE CONNECTION
30 02	CREATE PROGRAM
30 04	CREATE MAPSET
30 06	CREATE PARTITIONSET
30 08	CREATE TRANSACTION
30 1A	CREATE TRANCLASS
30 1C	CREATE TDQUEUE
30 1E	CREATE JOURNALMODEL
30 10	CREATE TERMINAL
30 12	CREATE SESSIONS
30 14	CREATE FILE
30 16	CREATE LSRPOOL
30 18	CREATE PARTNER
30 20	CREATE DB2CONN

Code	Command
30 22	CREATE DB2ENTRY
30 24	CREATE DB2TRAN
30 26	CREATE PROCESSTYPE
30 28	CREATE TSMODEL
30 2A	CREATE ENQMODEL
30 2C	CREATE REQUESTMODEL
30 2E	CREATE DOCTEMPLATE
30 30	CREATE TCPIPSERVICE
30 32	CREATE CORBASERVER
30 34	CREATE DJAR
30 36	CREATE URIMAP
30 38	CREATE PIPELINE
30 3A	CREATE WEBSERVICE
30 3C	CREATE IPCONN
30 3E	CREATE LIBRARY
30 40	CREATE BUNDLE
30 42	CREATE ATOMSERVICE
30 44	CREATE MQCONN
30 46	CREATE JVMSERVER
3A 02	INQUIRE RRMS
42 02	INQUIRE AUTINSTMODEL
42 10	DISCARD AUTINSTMODEL
44 02	INQUIRE PARTNER
44 10	DISCARD PARTNER
46 02	INQUIRE PROFILE
46 10	DISCARD PROFILE
4C 02	INQUIRE FILE
4C 04	SET FILE
4C 10	DISCARD FILE
4E 02	INQUIRE PROGRAM
4E 04	SET PROGRAM
4E 10	DISCARD PROGRAM
50 02	INQUIRE TRANSACTION
50 04	SET TRANSACTION
50 10	DISCARD TRANSACTION
52 02	INQUIRE TERMINAL
52 04	SET TERMINAL
52 06	INQUIRE NETNAME
52 08	SET NETNAME
52 10	DISCARD TERMINAL
52 12	INQUIRE TERMINAL
52 14	SET TERMINAL
52 16	INQUIRE NETNAME
54 02	INQUIRE SYSTEM
54 04	SET SYSTEM
58 02	INQUIRE CONNECTION
58 04	SET CONNECTION
58 06	PERFORM ENDAFFINITY
58 10	DISCARD CONNECTION
5A 02	INQUIRE MODENAME
5A 04	SET MODENAME
5C 02	INQUIRE TDQUEUE
5C 04	SET TDQUEUE
5C 10	DISCARD TDQUEUE

Code	Command
5E 02	INQUIRE TASK
5E 04	SET TASK
5E 08	INQUIRE STORAGE
5E 1A	INQUIRE TRANCLASS
5E 1C	SET TRANCLASS
5E 12	INQUIRE TCLASS
5E 14	SET TCLASS
5E 18	DISCARD TRANCLASS
5E 42	INQUIRE SUBPOOL
60 02	INQUIRE JOURNALNUM
60 04	SET JOURNALNUM
60 10	DISCARD JOURNALNAME
60 12	INQUIRE JOURNALNAME
60 14	SET JOURNALNAME
62 02	INQUIRE VOLUME
62 04	SET VOLUME
64 02	PERFORM SECURITY
66 02	INQUIRE DUMPDS
66 04	SET DUMPDS
66 12	INQUIRE TRANDUMPCODE
66 14	SET TRANDUMPCODE
66 22	INQUIRE SYSDUMPCODE
66 24	SET SYSDUMPCODE
68 02	INQUIRE VTAM
68 04	SET VTAM
68 12	INQUIRE AUTOINSTALL
68 14	SET AUTOINSTALL
68 22	INQUIRE DELETSHIPED
68 24	SET DELETSHIPED
68 26	PERFORM DELETSHIPED
6E 02	INQUIRE IRC
6E 04	SET IRC
70 02	INQUIRE STATISTICS
70 04	SET STATISTICS
70 06	PERFORM STATISTICS
70 08	COLLECT STATISTICS
70 12	INQUIRE MONITOR
70 14	SET MONITOR
70 26	EXTRACT STATISTICS
72 02	PERFORM RESETTIME
76 02	PERFORM SHUTDOWN
78 02	INQUIRE TRACEDEST
78 04	SET TRACEDEST
78 12	INQUIRE TRACEFLAG
78 14	SET TRACEFLAG
78 22	INQUIRE TRACETYPE
78 24	SET TRACETYPE
7A 02	INQUIRE DSNAME
7A 04	SET DSNAME
7C 02	INQUIRE EXCI
7E 04	PERFORM DUMP
80 02	INQUIRE TSQUEUE
80 04	SET TSQUEUE
80 12	INQUIRE TSQNAME

Code	Command
80 14	SET TSQNAME
80 1A	INQUIRE TSPOOL
80 22	INQUIRE TSMODEL
80 30	DISCARD TSMODEL
80 32	INQUIRE TEMPSTORAGE
80 34	SET TEMPSTORAGE
86 02	ACQUIRE TERMINAL
88 02	INQUIRE EXITPROGRAM
8A 02	INQUIRE REQID
90 02	INQUIRE UOW
90 04	SET UOW
90 22	INQUIRE ENQ
90 22	INQUIRE UOWENQ
90 42	INQUIRE UOWLINK
90 44	SET UOWLINK
90 62	INQUIRE UOWDSNFAIL
90 82	INQUIRE ENQMODEL
90 84	SET ENQMODEL
90 90	DISCARD ENQMODEL
92 02	INQUIRE JOURNALMODEL
92 10	DISCARD JOURNALMODEL
92 12	INQUIRE STREAMNAME
94 02	INQUIRE DB2CONN
94 04	SET DB2CONN
94 10	DISCARD DB2CONN
94 22	INQUIRE DB2ENTRY
94 24	SET DB2ENTRY
94 30	DISCARD DB2ENTRY
94 42	INQUIRE DB2TRAN
94 44	SET DB2TRAN
94 50	DISCARD DB2TRAN
96 02	INQUIRE PROCESSTYPE
96 04	SET PROCESSTYPE
96 10	DISCARD PROCESSTYPE
98 02	INQUIRE CFDTPOOL
9A 02	INQUIRE REQUESTMODEL
9A 10	DISCARD REQUESTMODEL
9C 02	INQUIRE TCPIPSERVICE
9C 04	SET TCPIPSERVICE
9C 10	DISCARD TCPIPSERVICE
9C 12	INQUIRE TCPIP
9C 14	SET TCPIP
9C 22	INQUIRE WEB
9C 24	SET WEB
9E 02	INQUIRE DOCTEMPLATE
9E 10	DISCARD DOCTEMPLATE
A2 02	CSD ADD
A2 04	CSD ALTER
A2 06	CSD APPEND
A2 08	CSD COPY
A2 0A	CSD DEFINE
A2 0C	CSD DELETE
A2 0E	CSD INSTALL
A2 10	CSD LOCK

Code	Command
A2 12	CSD REMOVE
A2 14	CSD RENAME
A2 16	CSD UNLOCK
A2 18	CSD USERDEFINE
A2 1A	CSD INQUIREGROUP
A2 1C	CSD INQUIRELIST
A2 1E	CSD INQUIRERSRCE
A2 20	CSD GETNEXTGROUP
A2 22	CSD GETNEXTLIST
A2 24	CSD GETNEXTSRCE
A2 26	CSD STARTBRGROUP
A2 28	CSD STARTBRLIST
A2 2A	CSD STARTBRRSRCE
A2 2C	CSD ENDBRGROUP
A2 2E	CSD ENDBRLIST
A2 30	CSD ENDBRRSRCE
A2 32	CSD DISCONNECT
B0 02	INQUIRE JVMPOOL
B0 04	SET JVMPOOL
B0 10	DISCARD JVMSERVER
B0 12	INQUIRE JVMPROFILE
B0 22	INQUIRE CLASSCACHE
B0 24	SET CLASSCACHE
B0 26	PERFORM CLASSCACHE
B0 32	INQUIRE JVM
B0 42	INQUIRE JVMSERVER
B0 44	SET JVMSERVER
B2 02	INQUIRE CORBASERVER
B2 04	SET CORBASERVER
B2 06	PERFORM CORBASERVER
B2 10	DISCARD CORBASERVER
B2 22	INQUIRE DJAR
B2 24	PERFORM DJAR
B2 30	DISCARD DJAR
B2 42	INQUIRE BEAN
B4 02	INQUIRE BRFACILITY
B4 04	SET BRFACILITY
B6 02	INQUIRE DISPATCHER
B6 04	SET DISPATCHER
B6 12	INQUIRE MVSTCB
B8 02	INQUIRE WORKREQUEST
B8 04	SET WORKREQUEST
BC 02	INQUIRE PIPELINE
BC 04	SET PIPELINE
BC 06	PERFORM PIPELINE
BC 10	DISCARD PIPELINE
BC 22	INQUIRE WEBSERVICE
BC 24	SET WEBSERVICE
BC 30	DISCARD WEBSERVICE
BE 02	INQUIRE URIMAP
BE 04	SET URIMAP
BE 10	DISCARD URIMAP
BE 12	INQUIRE HOST
BE 14	SET HOST

Code	Command
C2 02	INQUIRE IPCONN
C2 04	SET IPCONN
C2 10	DISCARD IPCONN
C2 12	INQUIRE IPFACILITY
C4 02	INQUIRE ASSOCIATION
C6 02	INQUIRE LIBRARY
C6 04	SET LIBRARY
C6 10	DISCARD LIBRARY
C8 02	INQUIRE BUNDLE
C8 04	SET BUNDLE
C8 12	INQUIRE BUNDLEPART
C8 10	DISCARD BUNDLE
CA 02	INQUIRE EVENTBINDING
CA 04	SET EVENTBINDING
CA 12	INQUIRE EVENTPROCESS
CA 14	SET EVENTPROCESS
CA 22	INQUIRE CAPTURESPEC
CC 02	INQUIRE ATOMSERVICE
CC 04	SET ATOMSERVICE
CC 10	DISCARD ATOMSERVICE
CE 02	INQUIRE MQCONN
CE 04	SET MQCONN
CE 10	DISCARD MQCONN
CE 12	INQUIRE MQINI
D0 02	INQUIRE XMLTRANSFORM
D0 04	SET XMLTRANSFORM

Appendix C. RESP2 values for EXEC CICS CREATE and EXEC CICS CSD commands

The **EXEC CICS CREATE** command and the **EXEC CICS CSD** commands **DEFINE**, **ALTER**, **USERDEFINE**, and **INSTALL** issue RESP2 values, each of which is associated with a CICS message. Most of these messages are written to transient data queue CSMT.

The RESP2 values and the corresponding message numbers are shown in Table 8 below. For this command, the fullword EIBRESP2 field is regarded as a structure containing two halfwords. The low-order halfword always contains an error number. The high-order halfword sometimes contains another number to help you to identify the error. Sometimes this number is the offset *n* in the ATTRIBUTES string at which the error was detected. Sometimes it is the keyword number *k* for which the error was detected. For a list of the keyword numbers, see Table 9 on page 845.

Table 8. RESP2 values corresponding to messages

RESP2	Msgid	Description or message
Codes caused by syntactical errors		
<i>n</i> ,400	DFHCA5211	A misplaced delimiter occurs in ATTRIBUTES. The invalid delimiter is at offset <i>n</i> in the ATTRIBUTES string.
<i>n</i> ,401	DFHCA5204	A keyword specified within ATTRIBUTES is invalid. The invalid keyword is at offset <i>n</i> in the ATTRIBUTES string.
<i>n</i> ,402	DFHCA5212, DFHCA5213	A keyword within ATTRIBUTES cannot be uniquely identified from its abbreviation. The invalid keyword is at offset <i>n</i> in the ATTRIBUTES string.
<i>k</i> ,403	DFHCA5501	A required keyword is omitted. The omitted keyword has code <i>k</i> in Table 9 on page 845.
404	DFHCA5529	A required keyword is omitted. The omitted keyword must be selected from two mutually exclusive keywords, as specified in the associated message.
<i>k</i> ,405	DFHCA5504	One specified keyword requires another one to be specified. The omitted keyword has code <i>k</i> in Table 9 on page 845.
<i>k</i> ,406	DFHCA5206	A keyword occurs more than once within ATTRIBUTES. The duplicate keyword has code <i>k</i> in Table 9 on page 845.
<i>k</i> ,407	DFHCA5503 DFHCA5506	Conflicting keywords are specified. The keyword causing the conflict has code <i>k</i> in Table 9 on page 845.
<i>k</i> ,410	DFHCA5210 DFHCA5519 DFHCA5521 DFHCA5522 DFHCA5526 DFHCA5528 DFHCA5532 DFHCA5547	An invalid operand is supplied for a keyword within ATTRIBUTES. The keyword in error has code <i>k</i> in Table 9 on page 845.
<i>k</i> ,410	DFHCA5224	The argument value is outside the valid range for keyword. The keyword in error has code <i>k</i> in Table 9 on page 845.
<i>k</i> ,410	DFHCA5542	Length of Remoteprefix and length of Prefix must be the same. The keyword in error has code <i>k</i> in Table 9 on page 845.
<i>k</i> ,410	DFHCA5543	Generics must be in the same place in the Prefix and in the Remoteprefix. The keyword in error has code <i>k</i> in Table 9 on page 845.
<i>k</i> ,411	DFHCA5207	An operand is supplied for a keyword that does not need one. The keyword in error has code <i>k</i> in Table 9 on page 845.

Table 8. RESP2 values corresponding to messages (continued)

RESP2	Msgid	Description or message
<i>k</i> ,412	DFHCA5205	A required operand for a keyword within ATTRIBUTES is omitted. The keyword in error has code <i>k</i> in Table 9 on page 845.
<i>k</i> ,412	DFHCA5544	The value must be specified as generic because a previous value is generic. The keyword in error has code <i>k</i> in Table 9 on page 845.
<i>k</i> ,413	DFHCA5517	The operands of two or more keywords conflict with one another. The first conflicting keyword detected has code <i>k</i> in Table 9 on page 845.
<i>k</i> ,414	DFHCA5507	The value of the operand of a keyword within ATTRIBUTES is too small. The keyword in error has code <i>k</i> in Table 9 on page 845.
<i>k</i> ,415	DFHCA5513	In the pair of values specified as the operand of a keyword within ATTRIBUTES, the second value must not exceed the first. The keyword in error has code <i>k</i> in Table 9 on page 845.
<i>k</i> ,416	DFHCA5509	An invalid operand is supplied for a keyword within ATTRIBUTES. The value of the operand must be different from the name of the resource. The keyword in error has code <i>k</i> in Table 9 on page 845.
417	DFHCA4884	The specified resource cannot be created with this command.
	DFHCA5523	The resource name is reserved for CICS use.
	DFHCA5535	
418	DFHCA5527	CICS internal programs (whose names begin with DFH) cannot be given attributes that specify remote execution.
<i>k</i> ,419	DFHCA5217	A closing parenthesis has been omitted from a DESCRIPTION keyword within ATTRIBUTES. The keyword in error (DESCRIPTION) has code <i>k</i> in Table 9 on page 845.
420	DFHCA5508	PROTECTNUM must be less than or equal to THREADLIMIT, or COMTHREADLIM must be less than or equal to TCBLIMIT, or THREADLIMIT must be less than or equal to TCBLIMIT.
421	DFHCA5544	Value must be specified as generic because a previous value is generic.
422	DFHCA5546	Incorrect TYPE specified for REQUESTMODEL.
423	DFHCA5548	The option is invalid for a request model from an earlier release of CICS.
424	DFHCA5549	The values specified for the two attributes must not be the same.
425	DFHCA5551	<i>keyword1</i> cannot be specified as generic unless <i>keyword2</i> is also generic. See the associated message to determine which keywords are in error.
426	DFHCA5552	The specified CIPHERS attribute has an invalid value.
427	DFHCA5553	The specified attribute field cannot contain a character as shown.
428	DFHCA5555	There must be at least one <i>attribute</i> present on resource definition.
429	DFHCA5556	The resource name starts with the reserved letter "C" or reserved letters "DFH" or "EYU".
	DFHAC5539	
430	DFHCA5557	The resource name used is a reserved name.
431	DFHCA5560	The HOST attribute contains a port number and a different, non-zero PORT attribute has also been specified.

Codes caused by errors deleting existing resources

Table 8. RESP2 values corresponding to messages (continued)

RESP2	Msgid	Description or message
500	DFHAM4803 DFHAM4834 DFHAM4836 DFHAM4842 DFHAM4896 DFHCA4940 DFHCA4803 DFHCA4834 DFHCA4836 DFHCA4842 DFHCA4896 DFHCA4940 DFHZC5913	Install failed because the resource is currently in use.
500	DFHAM4834 DFHAM4838 DFHCA4834 DFHCA4838	Install failed because the resource is not disabled.
500	DFHAM4853 DFHCA4853	Install failed because there is another DB2TRAN installed that specifies the same transaction ID.
500	DFHAM4874 DFHCA4874	Install failed because the attribute exists.
500	DFHAM4894 DFHCA4894	Install failed because the resource is not disabled.
500	DFHAM4903 DFHCA4903	Install failed because the service is open.
500	DFHAM4940 DFHCA4940	MQCONN install failed because an MQCONN is already installed.
501	DFHAM4841 DFHCA4841	Install failed because definition of <i>restype resname</i> is in use by task no. <i>taskno</i> (transaction id. <i>tranid</i>).
501	DFHZC5980	Resource <i>resource</i> is in use by task <i>taskid</i> Transaction <i>tranid</i> .
502	DFHZC6304	Deletion of remote terminal <i>termid</i> failed because it is in use by another transaction.
503	DFHZC5915	Deletion of <i>restype id</i> failed. It must be set out of service.
504	DFHAM4899 DFHCA4899 DFHZC5998	Install specified a resource that cannot be replaced.
505	DFHZC5916	Deletion of terminal <i>termid</i> failed. It has pending DFHZCP activity.
505	DFHZC5918	Deletion of terminal <i>termid</i> Console <i>consname</i> failed. It has pending DFHZCP activity.
506	DFHZC5914	Deletion of terminal <i>termid</i> found another deletion of it in progress.
506	DFHZC5937	Deletion of modename <i>modename</i> found another deletion of it in progress.
507	DFHZC5902	Deletion of terminal <i>termid</i> failed. BMS Paging session still active.
508	DFHZC5917	Deletion of terminal <i>termid</i> failed. Error message writer still active.
509	DFHZC5904	Deletion of terminal <i>termid</i> failed. CEDF is still active.
510	DFHZC5941	Install for terminal <i>termid</i> failed. Console <i>consname</i> has a conversation outstanding.
511	DFHZC5907	Deletion of remote shipped terminal failed for connection <i>cccc</i> .
512	DFHZC5925	Deletion of connection <i>cccc</i> failed. Its AID-Chains are not empty.
513	DFHZC5929	Deletion of connection <i>cccc</i> failed. It is in use by <i>n</i> indirect connections.

Table 8. RESP2 values corresponding to messages (continued)

RESP2	Msgid	Description or message
514	DFHZC5938	Deletion of modename <i>modename</i> failed. Unable to delete sessions.
515	DFHZC5951	Deletion of connection <i>ssss</i> failed. Unable to delete sessions.
516	DFHZC5945	Deletion of sessions <i>ssss</i> failed. Connection <i>cccc</i> is defined to IRC.
517	DFHZC5952	Deletion of terminal <i>termid</i> failed. It must be SET RELEASED.
518	DFHZC5969	Deletion of dependent modenames failed for connection <i>modename</i> .
519	DFHZC5974	Deletion of pool <i>pppp</i> failed. Unable to delete pool entries.
520	DFHZC5979	Deletion of pool <i>pppp</i> failed. It still has session <i>termid</i> .
520	DFHZC5982	Deletion of pool <i>pppp</i> failed. Pool entry is in use for <i>termid</i> .
521	DFHZC5958	Install failed for <i>xxxx</i> . This is the name of the local system, which must not be replaced.
522	DFHZC5940	Install for terminal <i>termid</i> failed. Error console cannot be deleted.
523	DFHZC5989	Deletion of resource <i>resource</i> failed. Remote deletion in connection <i>cccc</i> failed.
524	DFHZC5943	MRO connection <i>conname</i> cannot be deleted because IRC is open.
525	DFHAM4807	LSRPOOL install failed because its MAXKEYLENGTH is less than 22 which is incorrect for use by the CSD.
Codes caused by errors in installing the new resource		
600	DFHTO6000	The definition for TERMINAL <i>termdef</i> refers to an undefined TYPETERM <i>termtype</i> .
600	DFHTO6001	The definition for pooled TERMINAL <i>termdef</i> refers to an undefined TYPETERM <i>termtype</i> .
601	DFHAM4910 DFHCA4910	The install failed because the member was not found in the partitioned data set.
601	DFHTO6002	The definition for SESSIONs <i>sesdef</i> refers to an undefined CONNECTION <i>condef</i> .
601	DFHZC5911	Install for resource <i>resource</i> failed. Connection <i>cccc</i> not found.
601	DFHZC5932	Install for modename <i>modename</i> failed. Connection <i>cccc</i> not found.
602	DFHZC5962	Install for resource <i>resource</i> failed. Modename parameter not found.
603	DFHZC5906	Install failed because <i>xxxx</i> is not a permitted value for a terminal or connection name.
604	DFHZC5933	Install for modename <i>modename</i> failed. Connection <i>cccc</i> is not valid here.
605	DFHAM4889 DFHCA4889	Install of resource failed because an <i>attribute</i> is invalid.
606	DFHAM4890 DFHCA4890	Install of TDQUEUE <i>tdqname</i> failed because the TYPE has not been specified.
607	DFHAM4870 DFHCA4870	Install failed for program <i>progrname</i> - language RPG is not supported under MVS.
608	DFHAM4832 DFHCA4832	Unable to open TDQUEUE <i>tdqname</i> because the DFHINTRA data set is not open.
608	DFHAM4909 DFHCA4909	The install failed because the DDNAME was not found.
609	DFHAM4908 DFHCA4908	The installation failed because the templatenamename exists.
610	DFHAM4905 DFHCA4905	The installation failed because the option is unavailable in this system.
611	DFHAM4901 DFHCA4901	The installation of a REQUESTMODEL definition failed because a duplicate pattern exists.

Table 8. RESP2 values corresponding to messages (continued)

RESP2	Msgid	Description or message
612	DFHAM4920 DFHCA4920	The installation of a BUNDLE failed because it is a duplicate.
612	DFHCA4920	The installation of a resource failed because it exists.
613	DFHAM4921 DFHCA4921	The installation of a CORBASERVER or DJAR definition failed because an invalid JNDIPrefix, Shelf, or Certificate value has been specified.
614	DFHAM4923 DFHCA4923	The installation of a DJAR definition failed because the specified CorbaServer does not exist.
615	DFHAM4926 DFHCA4926	The installation of a CORBASERVER or DJAR definition failed because the resource resolution transaction CEJR could not attach.
616	DFHAM4927 DFHCA4927	The installation of a DJAR definition failed because its HFSFILE is a duplicate of one that exists.
617	DFHAM4926 DFHCA4926	The installation of a DJAR definition failed because the specified CORBASERVER is not in a valid state.
618	DFHAM4902 DFHCA4902	The installation of a CORBASERVER definition or of a REQUESTMODEL definition failed because it is not a valid definition for this CICS level.
619	DFHAM4925 DFHCA4925	The installation of a CORBASERVER failed because one of its associated TCPIP SERVICES was not installed.
620	DFHAM4912 DFHCA4912	The installation of the resource failed because the attribute specified is obsolete.
620	DFHZC5912	Install for terminal <i>termid</i> failed. It is incompatible with connection <i>cccc</i> .
620	DFHZC5949	Install for sessions <i>ssss</i> failed. It is incompatible with connection <i>cccc</i> .
621	DFHZC5900	System <i>sysid</i> has shipped definitions but connection <i>cccc</i> is not known to this system.
622	DFHZC5921	Install of terminal <i>termid</i> failed. z/OS Communications Server support not loaded.
622	DFHZC5988	Install for resource <i>resource</i> failed. z/OS Communications Server support not generated.
623	DFHZC5909	Install of resource <i>resource</i> failed. Call to DFHIRP <i>irp_function Return_code</i> did not succeed, See DFHIRSDS for return code.
624	DFHZC5931	Install for modename <i>modename</i> failed. Maximum number of APPC sessions would have been exceeded.
625	DFHAM4929 DFHCA4929	A resource was not installed because of conflicting attributes.
625	DFHZC5973	Install for sessions <i>ssss</i> failed. Max session-count reached for modename <i>modename</i> .
626	DFHAM4930 DFHCA4930	A URIMAP resource was not installed as it maps the same URI as an existing installed URIMAP.
626	DFHZC5955	SESNUMB greater than DLTHRED in the SIT (<i>nnnn</i>).
627	DFHAM4931 DFHCA4931	The installation of a WEBSERVICE failed because the associated WSBIND file or PIPELINE does not exist.
627	DFHAM4941 DFHCA4941	ATOMSERVICE install failed because the associated XML binding or Atom configuration file was not found.
627	DFHZC5934	Install for modename <i>modename</i> failed. Single-session connection <i>cccc</i> is already in use.
628	DFHAM4932 DFHCA4932	The installation of a WEBSERVICE failed because: <ul style="list-style-type: none"> • the associated files in z/OS UNIX are not correctly set up • or the associated PIPELINE is not correctly set up
628	DFHAM4942 DFHCA4942	ATOMSERVICE install failed because CICS was not authorized to access the associated XML binding or Atom configuration file.

Table 8. RESP2 values corresponding to messages (continued)

RESP2	Msgid	Description or message
628	DFHZC5936	Install for modename <i>modename</i> failed. Connection <i>cccc</i> has active modegroup <i>xxxx</i> .
628	DFHAM4946 DFHCA4946	The installation of BUNDLE <i>bundle</i> failed because CICS does not have authority to access the manifest found in the bundle root directory.
629	DFHAM4933 DFHCA4933	The installation of a PIPELINE failed because the file specified in the WSDIR attribute was inaccessible.
629	DFHAM4943 DFHCA4943	ATOMSERVICE install failed because the associated XML binding or Atom configuration file was invalid.
629	DFHZC5939	Install for <i>name</i> failed. Duplicate session- or modegroup-name for connection <i>sysid</i> .
630	DFHZC5946	Install for sessions <i>ssss</i> failed. Connection <i>cccc</i> is defined to IRC.
630	DFHAM4934 DFHCA4934	Urimap install failed due to the specified HOSTCODEPAGE and CHARACTERSET combination being invalid.
631	DFHZC5948	Install for sessions <i>ssss</i> failed. Connection <i>cccc</i> is not suitable for IRC.
631	DFHAM4918 DFHCA4918	Install failed because all cipher codes were rejected.
632	DFHAM4936 DFHCA4936	The installation of BUNDLE resource <i>resource</i> failed because it had an invalid manifest.
632	DFHZC5954	Install for resource <i>resource</i> failed. Unable to install sessions component.
633	DFHAM4937 DFHCA4937	The installation of BUNDLE resource <i>resource</i> failed because it had no manifest.
633	DFHZC5963	<i>operation</i> RUSIZE <i>xxxx</i> from terminal <i>termid</i> was greater than TYPETERM RUSIZE <i>yyyy</i> .
634	DFHZC5967	Install for modename <i>modename</i> failed. Unable to install sessions.
635	DFHAM4939 DFHCA4939	The installation of an ATOMSERVICE failed because of a configuration error.
635	DFHZC5968	Unable to install LU Services Manager for modename <i>modename</i> .
636	DFHZC5981	Pool <i>pppp</i> not found.
637	DFHZC5985	Install for resource <i>resource</i> failed. Unable to install connection component.
638	DFHTO6003	TERMINAL <i>termdef</i> specifies CONSNAME but refers to TYPETERM <i>termtype</i> which does not specify DEVICE=CONSOLE.
639	DFHTO6004	TERMINAL <i>termdef</i> does not specify CONSNAME but refers to TYPETERM <i>termtype</i> which specifies DEVICE=CONSOLE.
640	DFHTO6005	PRINTER or ALTPRINTER for TERMINAL <i>termdef</i> is invalid for the DEVICE specified in TYPETERM <i>termtype</i> .
641	DFHTO6006	PRINTERCOPY or ALTPRINTERCOPY for TERMINAL <i>termdef</i> is invalid for the DEVICE specified in TYPETERM <i>termtype</i> .
642	DFHTO6007	AUTINSTMODEL YES ONLY for TERMINAL <i>termdef</i> is invalid for the DEVICE specified in TYPETERM <i>termtype</i> .
643	DFHTO6008	
644	DFHTO6009	The definition for SESSIONs <i>sesdef</i> refers to CONNECTION <i>condef</i> which specifies a different PROTOCOL.
645	DFHTO6010	The definition for SESSIONs <i>sesdef</i> must specify PROTOCOL LU61 as it refers to an MRO CONNECTION <i>condef</i> .
646	DFHTO6011	SESSIONs <i>sesdef</i> must specify both SENDCOUNT and RECEIVECOUNT as it refers to an MRO CONNECTION <i>condef</i> .

Table 8. RESP2 values corresponding to messages (continued)

RESP2	Msgid	Description or message
647	DFHTO6013	No SESSIONs definition refers to CONNECTION <i>condef</i> .
648	DFHTO6014	POOL is required for TERMINAL <i>termdef</i> as it refers to TYPETERM <i>typedef</i> which specifies SESSIONTYPE=PIPELINE.
649	DFHTO6015	TRANSACTION for TERMINAL <i>termdef</i> is invalid for the DEVICE specified in TYPETERM <i>typedef</i> .
650	DFHTO6016	The MRO CONNECTION <i>condef</i> is referenced by more than one SESSIONs definition, including <i>sesdef</i> .
651	DFHTO6017	REMOTESYSTEM for TERMINAL <i>termid</i> is invalid for the DEVICE specified in TYPETERM <i>typeterm</i> .
652	DFHTO6018	TERMINAL <i>termid</i> refers to TYPETERM <i>typeterm</i> which has an invalid ALTSCREEN.
653	DFHTO6020	SESSIONS <i>sesdef</i> refers to single-session CONNECTION <i>condef</i> but has an invalid MAXIMUM option specified.
654	DFHTO6023	Connection definition @.BCH detected. Batch-shared database connections are not supported.
655	DFHTO6025	The definition for LU6.1 SESSIONs <i>sesdef</i> specifies a send or receive count with no prefix.
656	DFHZC6301	Install for <i>tttt</i> failed. Duplicate netname <i>netname</i> for resource <i>rrrr</i> found.
657	DFHZC6302	Install for connection <i>cccc</i> failed. Duplicate netname <i>netname</i> for resource <i>rrrr</i> found.
658	DFHZC6303	Install for <i>tttt</i> failed. Duplicate netname <i>netname</i> found.
659	DFHZC6334	Install for <i>tttt</i> failed. A session with the same name exists.
660	DFHZC6331	Install for connection <i>tttt</i> failed. Non-z/OS Communications Server terminal with same name exists.
660	DFHZC6332	Install for terminal <i>tttt</i> failed. Non-z/OS Communications Server terminal with same name exists.
661	DFHZC5950	Install for terminal <i>termid</i> failed. Console <i>consname</i> exists.
664	DFHZC6330	Install for <i>tttt</i> failed. LDCLIST parameter <i>ldclist</i> not found.
665	DFHZC6333	INSTALL for modename <i>modename</i> failed. Zero sessions specified.
666	DFHAM4833 DFHCA4833 DFHZC6315 DFHZC6361	Resource cannot be installed with specified user ID because of a security error.
667	DFHZC6362	Install for terminal <i>portname</i> with user ID <i>userid</i> failed because the preset userID has been revoked.
668	DFHZC6363	Install for terminal <i>portname</i> with user ID <i>userid</i> failed because the preset userID's group access has been revoked.
669	DFHZC6364	Install for terminal <i>portname</i> with user ID <i>userid</i> failed because the ESM returned an unrecognized response.
670	DFHZC6365	Install for terminal <i>portname</i> with user ID <i>userid</i> failed because the external security manager is inactive.
671	DFHZC6366	Install for terminal <i>portname</i> with user ID <i>userid</i> failed because the userID is not authorized to access this CICS system.
672	DFHZC6367	Install for terminal <i>termid</i> with user ID <i>userid</i> failed because the SECLABEL check failed.
673	DFHZC6368	Install for terminal <i>portname</i> with user ID <i>userid</i> failed because the external security manager is quiesced.
674	DFHZC6369	Install for terminal <i>portname</i> failed because national language <i>langcode</i> is invalid.
675	DFHZC6370	Install for terminal <i>portname</i> failed because national language <i>langcode</i> is unavailable.
676	DFHZC6371	Install for terminal <i>portname</i> with userID <i>userid</i> failed because the userID is not authorized to use this portname.

Table 8. RESP2 values corresponding to messages (continued)

RESP2	Msgid	Description or message
677	DFHZC5944	Install for <i>type(id)</i> has failed. It would make a loop of connection definitions.
679	DFHAM4837 DFHCA4837	Install of DB2ENTRY or DB2TRAN failed because DB2CONN not installed.
680	DFHAM4850 DFHCA4850	DB2TRAN not installed because refers to a DB2ENTRY that is not installed.
681	DFHAM4851 DFHCA4851	DB2CONN not installed because of a security error, or DB2ENTRY not installed because of a security error, or DB2TRAN not installed because of a security error.
681	DFHCA4851	LIBRARY not installed because of a security error.
682	DFHAM4895 DFHCA4895	The TST has not been assembled with the migrate option when defining CREATE TSMODEL.
683	DFHCA4817	A serious MVS Abend was encountered during install processing.
685	DFHCA4869	Single resource installation for this resource type is not allowed.
686	DFHAM4947 DFHCA4947	The installation of BUNDLE resource failed because an unexpected resource error occurred.
Codes caused by CICS internal logic errors		
900	DFHTO6012	The catalog data set is not available. RDO function is restricted.
901	DFHAM4872 DFHCA4872	Unable to connect to CICS catalog.
902	DFHAM4873 DFHCA4873	Unable to disconnect the CICS catalog.
903	DFHZC6209	Invalid ZC catalog request code <i>xxxx</i> .
904	DFHZC6212	Level mismatch with catalog record. DFHBS <i>xxx</i> .
905	DFHAM4898 DFHCA4898 DFHZC5901	Install failed because sufficient storage could not be obtained.
906	DFHZC6200	Could not obtain DWE storage.
907	DFHZC6203	Unable to obtain DWE action-list storage.
908	DFHZC6214	Unable to obtain recovery record storage.
950	DFHZC6202	Pattern <i>pattern</i> not valid for builder.
951	DFHZC6204	Illegal subpattern definition <i>pattern</i> .
952	DFHZC6205	Illegal subpattern definition <i>pattern</i> .
953	DFHZC6206	Pattern <i>pattern</i> not valid when destroying a resource.
954	DFHZC6207	Catalog key too long or zero. Pattern <i>pattern</i> .
955	DFHZC6213	Recovery record abandoned. Key is <i>key</i> .
956	DFHZC6341	Loop or ABEND has been detected in <i>inmodule</i> by module <i>bymodule</i> .

Table 9. Keywords associated with keyword numbers.

Notes:

- Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
- Keywords shown as obsolete are retained for cross-release compatibility.

Resource	Keyword number	Keyword
ATOMSERVICE	1	ATOMSERVICE
	6	DESCRIPTION
	7	CONFIGFILE
	8	BINDFILE
	9	RESOURCENAME
	97	ATOMTYPE
	98	STATUS
	99	RESOURCETYPE
BUNDLE	1	BUNDLE
	6	DESCRIPTION
	7	BUNDLEDIR
	8	BASESCOPE
	97	STATUS
CONNECTION	1	CONNECTION
	5	NETNAME
	6	INDSYS
	7	SECURITYNAME
	8	BINDPASSWORD (obsolete)
	10	REMOTESYSTEM
	11	REMOTENAME
	12	DESCRIPTION
	13	QUEUELIMIT
	14	MAXQTIME
	27	REMOTESYSNET
	97	INSERVICE
	98	AUTOCONNECT
	99	PROTOCOL
	100	ACCESSMETHOD
	101	SINGLESESS
	102	DATASTREAM
	103	RECORDFORMAT
	104	ATTACHSEC
	105	BINDSECURITY
106	CONNTYPE	
107	PSRECOVERY	
110	USEDFLTUSER	
111	XLNACTION	

Table 9. Keywords associated with keyword numbers (continued).

Notes:

- Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
- Keywords shown as obsolete are retained for cross-release compatibility.

Resource	Keyword number	Keyword
CORBASERVER	1	CORBASERVER
	5	JNDIPREFIX
	7	SHELF
	8	HOST
	10	SESSBEANTIME(DAYS)
	11	SESSBEANTIME(HOURS)
	12	SESSBEANTIME(MINS)
	13	CERTIFICATE
	15	UNAUTH
	17	CLIENTCERT
	20	SSLUNAUTH
	21	DJARDIR
	22	CIPHERS
	98	AUTOPUBLISH
DB2CONN	1	DB2CONN
	6	DESCRIPTION
	7	DB2ID
	8	MSGQUEUE1
	9	MSGQUEUE2
	10	MSGQUEUE3
	11	PURGECYCLE
	13	STATSQUEUE
	14	TCBLIMIT
	15	THREADLIMIT
	16	AUTHID
	17	PLAN
	18	PLANEXITNAME
	19	COMTHREADLIMIT
	20	COMAUTHID
	21	SIGNID
	25	DB2GROUPIX
	98	CONNECTERROR
	99	NONTERMREL
	100	STANDBYMODE
	101	THREADERROR
	102	ACCOUNTREC
	103	AUTHTYPE
	104	DROLLBACK
106	PRIORITY	
107	THREADWAIT	
108	COMAUTHTYPE	
	110	RESYNCMEMBER

Table 9. Keywords associated with keyword numbers (continued).

Notes:

- Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
- Keywords shown as obsolete are retained for cross-release compatibility.

Resource	Keyword number	Keyword
DB2ENTRY	1	DB2ENTRY
	6	DESCRIPTION
	8	TRANSID
	13	PROTECTNUM
	15	THREADLIMIT
	16	AUTHID
	17	PLAN
	18	PLANEXITNAME
	102	ACCOUNTREC
	103	AUTHTYPE
	104	DROLLBACK
DB2TRAN	106	PRIORITY
	107	THREADWAIT
	1	DB2TRAN
	6	DESCRIPTION
	8	TRANSID
DJAR	9	ENTRY
	1	DJAR
	5	CORBASERVER
DOCTEMPLATE	7	HFSFILE
	1	DOCTEMPLATE
	6	DESCRIPTION
	7	TEMPLATE
	8	FILE
	9	TSQUEUE
	10	TDQUEUE
	11	PROGRAM
	12	EXITPGM
	13	DDNAME
	14	MEMBERNAME
	17	HFSFILE
	ENQMODEL	99
100		TYPE
1		ENQMODEL
6		DESCRIPTION
7		ENQSCOPE
8		ENQNAME
	99	STATUS

Table 9. Keywords associated with keyword numbers (continued).

Notes:

- Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
- Keywords shown as obsolete are retained for cross-release compatibility.

Resource	Keyword number	Keyword
FILE	1	FILE
	5	RESSECNUM (obsolete)
	6	DSNAME
	7	RECORDSIZE
	8	KEYLENGTH
	9	JOURNAL
	10	REMOTESYSTEM
	11	REMOTENAME
	12	PASSWORD
	13	LSRPOOLID
	14	STRINGS
	15	DATABUFFERS
	16	INDEXBUFFERS
	17	FWDRECOVLOG
	18	DESCRIPTION
	19	NSRGROUP
	20	MAXNUMRECS
	21	CFDTPOOL
	22	TABLERNAME
	23	LSRPOOLNUM
	97	STATUS
	98	RECOVERY
	99	OPENTIME
	100	DISPOSITION
	101	ADD
	102	BROWSE
	103	DELETE
	104	READ
	105	UPDATE
	106	JNLSYNCREAD
	107	JNLSYNCWRITE
	108	JNLREAD
	109	JNLUPDATE
	110	JNLADD
111	DSNSHARING	
112	RECORDFORMAT	
113	TABLE	
114	BACKUPTYPE	
115	RLSACCESS	
116	READINTEG	
117	LOAD	
118	UPDATEMODEL	

Table 9. Keywords associated with keyword numbers (continued).

Notes:

- Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
- Keywords shown as obsolete are retained for cross-release compatibility.

Resource	Keyword number	Keyword
IPCONN	1	IPCONN
	6	DESCRIPTION
	7	APPLID
	8	TCPIPSERVICE
	9	PORT
	10	HOST
	11	RECEIVECOUNT
	12	SENDCOUNT
	13	QUEUELIMIT
	14	MAXQTIME
	15	NETWORKID
	97	INSERVICE
	98	AUTOCONNECT
	99	XLNACTION
	100	SSL
JOURNALMODEL	101	USERAUTH
	102	LINKAUTH
	103	IDPROP
	104	MIRRORLIFE
	1	JOURNALMODEL
	6	DESCRIPTION
	7	JOURNALNAME
	8	STREAMNAME
98	TYPE	
JVMSERVER	1	JVMSERVER
	6	DESCRIPTION
	7	JVMPROFILE
	8	LERUNOPTS
	9	THREADLIMIT
	97	STATUS

Table 9. Keywords associated with keyword numbers (continued).

Notes:

- Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
- Keywords shown as obsolete are retained for cross-release compatibility.

Resource	Keyword number	Keyword
LIBRARY	1	LIBRARY
	5	RANKING
	6	DESCRIPTION
	7	DSNAME01
	8	DSNAME02
	9	DSNAME03
	10	DSNAME04
	11	DSNAME05
	12	DSNAME06
	13	DSNAME07
	14	DSNAME08
	15	DSNAME09
	16	DSNAME10
	17	DSNAME11
	18	DSNAME12
	19	DSNAME13
	20	DSNAME14
	21	DSNAME15
	22	DSNAME16
		97
	98	CRITICAL

Table 9. Keywords associated with keyword numbers (continued).

Notes:

- Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
- Keywords shown as obsolete are retained for cross-release compatibility.

Resource	Keyword number	Keyword
LSRPOOL	1	LSRPOOL
	6	MAXKEYLENGTH
	7	SHARELIMITE
	8	STRINGS
	9	DATA512
	10	DATA1K
	11	DATA2K
	12	DATA4K
	13	DATA8K
	14	DATA12K
	15	DATA16K
	16	DATA20K
	17	DATA24K
	18	DATA28K
	19	DATA32K
	20	LSRPOOLID
	21	DESCRIPTION
	22	INDEX512
	23	INDEX1K
	24	INDEX2K
	25	INDEX4K
	26	INDEX8K
	27	INDEX12K
	28	INDEX16K
	29	INDEX20K
	30	INDEX24K
	31	INDEX28K
	32	INDEX32K
	33	HSDATA4K
	34	HSDATA8K
	35	HSDATA12K
	36	HSDATA16K
	37	HSDATA20K
	38	HSDATA24K
	39	HSDATA28K
	40	HSDATA32K
	41	HSINDEX4K
	42	HSINDEX8K
	43	HSINDEX12K
	44	HSINDEX16K
	45	HSINDEX20K
	46	HSINDEX24K
	47	HSINDEX28K
	48	HSINDEX32K
	49	LSRPOOLNUM

Table 9. Keywords associated with keyword numbers (continued).

Notes:

- Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
- Keywords shown as obsolete are retained for cross-release compatibility.

Resource	Keyword number	Keyword
MAPSET	1	MAPSET
	5	RSL (obsolete)
	6	DESCRIPTION
	97	STATUS
	100	RESIDENT
	101	USAGE
	102	USELPACOPY
MQCONN	1	MQCONN
	5	MQNAME
	6	DESCRIPTION
	7	INITQNAME
	97	RESYNCMEMBER
PARTITIONSET	1	PARTITIONSET
	5	RSL (obsolete)
	6	DESCRIPTION
	97	STATUS
	100	RESIDENT
	101	USAGE
	102	USELPACOPY
PARTNER	1	PARTNER
	5	NETNAME
	6	DESCRIPTION
	7	NETWORK
	8	PROFILE
	9	TPNAME
	10	XTPNAME
PIPELINE	1	PIPELINE
	5	CONFIGFILE
	6	DESCRIPTION
	8	SHELF
	9	WSDIR
	97	STATUS
PROCESSTYPE	1	PROCESSTYPE
	5	AUDITLOG
	6	DESCRIPTION
	7	FILE
	98	STATUS
	99	AUDITLEVEL
	100	USERRECORDS

Table 9. Keywords associated with keyword numbers (continued).

Notes:

- Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
- Keywords shown as obsolete are retained for cross-release compatibility.

Resource	Keyword number	Keyword
PROFILE	1	PROFILE
	5	MODENAME
	6	JOURNAL
	7	NEPCCLASS
	8	RTIMOUT
	9	DESCRIPTION
	10	FACILITYLIKE
	98	SCRNSIZE
	99	MSGJRNL
	100	MSGINTEG
	101	ONEWTE
	102	PROTECT (obsolete)
	103	DVSUPRT
	104	INBFMH
	105	RAQ
	106	LOGREC
	107	PRINTERCOMP
	108	CHAINCONTROL
	109	UCTRAN
PROGRAM	1	PROGRAM
	5	RSL (obsolete)
	6	DESCRIPTION
	7	REMOTESYSTEM
	8	REMOTENAME
	9	TRANSID
	10	JVMCLASS
	33	JVMPROFILE
	97	STATUS
	98	LANGUAGE
	99	RELOAD
	100	RESIDENT
	101	USAGE
	102	USELPACOPY
	103	CEDF
	104	DATALOCATION
	105	EXECKEY
	107	EXECUTIONSET
	108	DYNAMIC
	109	CONCURRENCY
	110	JVM
	111	HOTPOOL (obsolete)
112	API	

Table 9. Keywords associated with keyword numbers (continued).

Notes:

- Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
- Keywords shown as obsolete are retained for cross-release compatibility.

Resource	Keyword number	Keyword
REQUESTMODEL	1	REQUESTMODEL
	6	DESCRIPTION
	7	OMGMODULE (obsolete)
	8	OMGINTERFACE (obsolete)
	9	OMGOPERATION (obsolete)
	10	TRANSID
	11	CORBASERVER
	12	BEANNAME
	13	MODULE
	14	INTERFACE
	15	OPERATION
	97	TYPE
	98	INTFACETYPE
SESSIONS	1	SESSIONS
	5	CONNECTION
	6	SESSNAME
	7	NETNAMEQ
	8	MODENAME
	9	MAXIMUM
	11	RECEIVEPFX
	12	RECEIVECOUNT
	13	SENDPFX
	14	SENDCOUNT
	15	OPERID (obsolete)
	16	OPERPRIORITY (obsolete)
	17	OPERRSL (obsolete)
	18	OPERSECURITY (obsolete)
	19	USERID
	20	SENDSIZE
	21	RECEIVESIZE
	22	TRANSACTION (obsolete)
	23	SESSPRIORITY
	24	USERAREALEN
	25	IOAREALEN
	27	NEPCLASS
	28	DESCRIPTION
	97	INSERVICE (obsolete)
	98	AUTOCONNECT
	99	BUILDCHAIN
	100	PROTOCOL
	101	RELREQ
	102	DISCREQ
	103	RECOVOPTION
	104	RECOVNOTIFY (obsolete)

Table 9. Keywords associated with keyword numbers (continued).

Notes:

- Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
- Keywords shown as obsolete are retained for cross-release compatibility.

Resource	Keyword number	Keyword
TCPIPSERVICE	1	TCPIPSERVICE
	6	DESCRIPTION
	14	URM
	15	PORTNUMBER
	16	CIPHERS
	17	CERTIFICATE
	18	TRANSACTION
	19	BACKLOG
	20	REALM
	21	HOST
	22	TSQPREFIX
	23	IPADDRESS
	24	SOCKETCLOSE
	25	DNSGROUP
	27	MAXDATALEN
	28	MAXPERSIST
	104	STATUS
	105	SSL
	106	PROTOCOL
	109	AUTHENTICATE
110	GRPCRITICAL	
111	ATTACHSEC	
112	PRIVACY (obsolete)	

I

Table 9. Keywords associated with keyword numbers (continued).

Notes:

- Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
- Keywords shown as obsolete are retained for cross-release compatibility.

Resource	Keyword number	Keyword
TDQUEUE	1	TDQUEUE
	6	DESCRIPTION
	7	BLOCKSIZE
	8	DATABUFFERS
	9	DDNAME
	10	DSNAME
	11	RECORDSIZE
	12	FACILITYID
	13	TRANSID
	14	TRIGGERLEVEL
	15	USERID
	16	INDIRECTNAME
	17	REMOTENAME
	18	REMOTESYSTEM
	19	SYSOUTCLASS
	20	REMOTELength
	98	TYPE
	99	DISPOSITION
	100	ERROROPTION
	101	OPENTIME
	102	RECORDFORMAT
	103	BLOCKFORMAT
	104	REWIND
105	TYPEFILE	
106	ATIFACILITY	
107	RECOVSTATUS	
108	WAITACTION	
109	PRINTCONTROL	
110	WAIT	

Table 9. Keywords associated with keyword numbers (continued).

Notes:

- Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
- Keywords shown as obsolete are retained for cross-release compatibility.

Resource	Keyword number	Keyword
TERMINAL	1	TERMINAL
	6	AUTINSTNAME
	7	TYPETERM
	8	NETNAME
	9	CONSOLE (obsolete)
	10	REMOTESYSTEM
	11	REMOTENAME
	12	MODENAME
	13	PRINTER
	14	ALTPRINTER
	15	OPERID (obsolete)
	16	OPERPRIORITY (obsolete)
	17	OPERRSL (obsolete)
	18	OPERSECURITY (obsolete)
	19	USERID
	20	POOL
	21	TASKLIMIT
	22	TRANSACTION
	23	TERMPRIORITY
	26	SECURITYNAME
	27	BINDPASSWORD (obsolete)
	28	DESCRIPTION
	29	NATLANG
	30	CONSNAME
	33	REMOTESYSNET
	97	INSERVICE
	98	PRINTERCOPY
	99	ALTPRINCOPY
	100	AUTINSTMODEL
	102	ATTACHSEC
	103	BINDSECURITY
104	USEDFTUSER	
TRANCLASS	1	TRANCLASS
	5	MAXACTIVE
	6	DESCRIPTION
	7	PURGETHRESH

Table 9. Keywords associated with keyword numbers (continued).

Notes:

- Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
- Keywords shown as obsolete are retained for cross-release compatibility.

Resource	Keyword number	Keyword
TRANSACTION	1	TRANSACTION
	5	RSL (obsolete)
	6	PROGRAM
	7	TWASIZE
	8	PROFILE
	9	PARTITIONSET
	10	REMOTESYSTEM
	11	REMOTENAME
	12	PRIORITY
	13	TCLASS (obsolete)
	14	TASKREQ
	15	XTRANID
	16	DTIMOUT
	17	TRANSEC (obsolete)
	18	TRPROF
	19	PRIMEDSIZE (obsolete)
	20	ALIAS
	21	DESCRIPTION
	22	TPNAME
	23	XTPNAME
	24	TRANCLASS
	25	RUNAWAY
	26	WAITTIME
	29	BREXIT
	97	STATUS
	98	LOCALQ
	99	INDOUBT (obsolete)
	100	RESTART
	101	SPURGE
	102	TPURGE
	103	DUMP
	104	EXTSEC (obsolete)
	105	RESSEC
106	TRACE	
107	DYNAMIC	
108	CMDSEC	
109	TASKDATALOC	
110	TASKDATAKEY	
111	STORAGECLEAR	
112	SHUTDOWN	
113	ISOLATE	
114	CONFDATA	
115	WAIT	
116	ACTION	
117	ROUTABLE	

Table 9. Keywords associated with keyword numbers (continued).

Notes:

- Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
- Keywords shown as obsolete are retained for cross-release compatibility.

Resource	Keyword number	Keyword
TSMODEL	1	TSMODEL
	6	DESCRIPTION
	7	PREFIX
	8	POOLNAME
	9	REMOTESYSTEM
	10	REMOTEPREFIX
	11	XPREFIX
	12	XREMOTEPEFX
	13	EXPIRYINT
	99	LOCATION
	100	RECOVERY
	101	SECURITY

Table 9. Keywords associated with keyword numbers (continued).

Notes:

- Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
- Keywords shown as obsolete are retained for cross-release compatibility.

Resource	Keyword number	Keyword
TYPETERM	1	TYPETERM
	5	DEVICE
	6	TERMMODEL
	7	SESSIONTYPE
	9	LDCLIST
	10	DEFSCREEN
	12	ALTSCREEN
	14	CGCSGID
	16	SENDSIZE
	17	RECEIVESIZE
	18	LOGMODE
	19	PAGESIZE
	21	ALTPAGE
	23	ALTSUFFIX
	24	USERAREALEN
	25	IOAREALEN
	27	NEPCCLASS
	28	DESCRIPTION
	98	AUTOCONNECT
	99	SHIPPABLE
	100	APLKYBD
	101	APLTEXT
	102	AUDIBLEALARM
	103	COLOR
	104	COPY
	105	DUALCASEKYBD
	106	EXTENDEDDES
	107	HILIGHT
108	KATAKANA	
109	LIGHTPEN	
110	MSRCONTROL	
111	OBFORMAT	
112	PARTITIONS	
113	PRINTADAPTER	
114	PROGSYMBOLS	
115	VALIDATION	
116	FORMFEED	
117	HORIZFORM	
118	VERTICALFORM	
119	TEXTKYBD	
120	TEXTPRINT	
121	QUERY	
122	OUTLINE	
123	SOSI	
124	BACKTRANS	
125	ASCII	
126	BRACKET	
127	FMHPARM	
128	OBOPERID	

Table 9. Keywords associated with keyword numbers (continued).

Notes:

- Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
- Keywords shown as obsolete are retained for cross-release compatibility.

Resource	Keyword number	Keyword
URIMAP	1	URIMAP
	14	SOCKETCLOSE
	15	PORT
	16	HOST
	17	PATH
	21	TCPIPSERVICE
	23	TRANSACTION
	24	CONVERTER
	25	PROGRAM
	26	PIPELINE
	27	WEBSERVICE
	32	USERID
	33	CIPHERS
	34	CERTIFICATE
	35	MEDIATYPE
	36	CHARACTERSET
	37	HOSTCODEPAGE
	38	TEMPLATENAME
	39	HFSFILE
	40	LOCATION
97	STATUS	
98	USAGE	
99	SCHEME	
106	ANALYZER	
WEBSERVICE	1	WEBSERVICE
	5	PIPELINE
	6	DESCRIPTION
	7	WSBIND
	8	WSDLFILE
	97	VALIDATION

Appendix D. Threadsafe SPI commands

CICS provides threadsafe SPI commands for you to use in your programs.

Threadsafe SPI commands

DISCARD ATOMSERVICE
DISCARD BUNDLE
DISCARD DB2CONN
DISCARD DB2ENTRY
DISCARD DB2TRAN
DISCARD DOCTEMPLATE
DISCARD JVMSERVER
DISCARD MQCONN
DISCARD PIPELINE
DISCARD URIMAP
DISCARD WEBSERVICE
INQUIRE ASSOCIATION
INQUIRE ASSOCIATION LIST
INQUIRE ATOMSERVICE
INQUIRE BUNDLE
INQUIRE BUNDLEPART
| INQUIRE CAPDATAPRED
| INQUIRE CAPINFOSRCE
| INQUIRE CAPOPTPRED
| INQUIRE CAPTURESPEC
| INQUIRE CLASSCACHE
INQUIRE DB2CONN
INQUIRE DB2ENTRY
INQUIRE DB2TRAN
INQUIRE DOCTEMPLATE
| INQUIRE EPADAPTER
INQUIRE EVENTBINDING
INQUIRE EVENTPROCESS
INQUIRE EXITPROGRAM
INQUIRE FILE
INQUIRE HOST
INQUIRE IPCONN
| INQUIRE JVM
| INQUIRE JVMPOOL
| INQUIRE JVMPROFILE
INQUIRE JVMSERVER
INQUIRE LIBRARY
INQUIRE MQCONN
INQUIRE MQINI
| INQUIRE OSGIBUNDLE
| INQUIRE OSGISERVICE
INQUIRE PIPELINE
INQUIRE TASK
| INQUIRE TEMPSTORAGE
INQUIRE URIMAP
INQUIRE WEB
INQUIRE WEBSERVICE
INQUIRE WORKREQUEST

Threadsafe SPI commands

INQUIRE XMLTRANSFORM
|
PERFORM CLASSCACHE
|
PERFORM JVMPOOL
PERFORM PIPELINE
|
RESYNC ENTRYNAME
SET ATOMSERVICE
SET BUNDLE
|
SET CLASSCACHE
SET DB2CONN
SET DB2ENTRY
SET DB2TRAN
SET DOCTEMPLATE
|
SET EPADAPTER
SET EVENTBINDING
SET EVENTPROCESS
SET HOST
SET IPCONN
|
SET JVMPOOL
SET JVMSERVER
SET LIBRARY
SET MQCONN
SET PIPELINE
|
SET TEMPSTORAGE
SET URIMAP
SET WEB
SET WEBSERVICE
SET WORKREQUEST
SET XMLTRANSFORM

For an introduction to the concept of threadsafe commands see “CICS threadsafe commands in the SPI” on page 29.

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 want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Bibliography

CICS books for CICS Transaction Server for z/OS

General

CICS Transaction Server for z/OS Program Directory, GI13-0565
CICS Transaction Server for z/OS What's New, GC34-7192
CICS Transaction Server for z/OS Upgrading from CICS TS Version 3.1, GC34-7188
CICS Transaction Server for z/OS Upgrading from CICS TS Version 3.2, GC34-7189
CICS Transaction Server for z/OS Upgrading from CICS TS Version 4.1, GC34-7190
CICS Transaction Server for z/OS Installation Guide, GC34-7171

Access to CICS

CICS Internet Guide, SC34-7173
CICS Web Services Guide, SC34-7191

Administration

CICS System Definition Guide, SC34-7185
CICS Customization Guide, SC34-7161
CICS Resource Definition Guide, SC34-7181
CICS Operations and Utilities Guide, SC34-7213
CICS RACF Security Guide, SC34-7179
CICS Supplied Transactions, SC34-7184

Programming

CICS Application Programming Guide, SC34-7158
CICS Application Programming Reference, SC34-7159
CICS System Programming Reference, SC34-7186
CICS Front End Programming Interface User's Guide, SC34-7169
CICS C++ OO Class Libraries, SC34-7162
CICS Distributed Transaction Programming Guide, SC34-7167
CICS Business Transaction Services, SC34-7160
Java Applications in CICS, SC34-7174

Diagnosis

CICS Problem Determination Guide, GC34-7178
CICS Performance Guide, SC34-7177
CICS Messages and Codes Vol 1, GC34-7175
CICS Messages and Codes Vol 2, GC34-7176
CICS Diagnosis Reference, GC34-7166
CICS Recovery and Restart Guide, SC34-7180
CICS Data Areas, GC34-7163
CICS Trace Entries, SC34-7187
CICS Debugging Tools Interfaces Reference, GC34-7165

Communication

CICS Intercommunication Guide, SC34-7172
CICS External Interfaces Guide, SC34-7168

Databases

CICS DB2 Guide, SC34-7164
CICS IMS Database Control Guide, SC34-7170

CICSplex SM books for CICS Transaction Server for z/OS

General

CICSplex SM Concepts and Planning, SC34-7196
CICSplex SM Web User Interface Guide, SC34-7214

Administration and Management

CICSplex SM Administration, SC34-7193
CICSplex SM Operations Views Reference, SC34-7202
CICSplex SM Monitor Views Reference, SC34-7200
CICSplex SM Managing Workloads, SC34-7199
CICSplex SM Managing Resource Usage, SC34-7198
CICSplex SM Managing Business Applications, SC34-7197

Programming

CICSplex SM Application Programming Guide, SC34-7194
CICSplex SM Application Programming Reference, SC34-7195

Diagnosis

CICSplex SM Resource Tables Reference Vol 1, SC34-7204
CICSplex SM Resource Tables Reference Vol 2, SC34-7205
CICSplex SM Messages and Codes, GC34-7201
CICSplex SM Problem Determination, GC34-7203

Other CICS publications

The following publications contain further information about CICS, but are not provided as part of CICS Transaction Server for z/OS, Version 4 Release 2.

Designing and Programming CICS Applications, SR23-9692
CICS Application Migration Aid Guide, SC33-0768
CICS Family: API Structure, SC33-1007
CICS Family: Client/Server Programming, SC33-1435
CICS Family: Interproduct Communication, SC34-6853
CICS Family: Communicating from CICS on System/390, SC34-6854
CICS Transaction Gateway for z/OS Administration, SC34-5528
CICS Family: General Information, GC33-0155
CICS 4.1 Sample Applications Guide, SC33-1173
CICS/ESA 3.3 XRF Guide, SC33-0661

Other IBM publications

The following publications contain information about related IBM products.

Resource Access Control Facility (RACF): Security Administrator's Guide, SC28-1340
IMS Utilities Reference, SC26-4284
IMS Messages and Code, SC26-4290
IMS™ System Administration Guide, SC26-4278
IMS Operations Guide, SC26-4287
An Introduction to the IBM 3270 Information Display System, GA27-2739
Systems Network Architecture—LU6.2 Reference: Peer Protocols, SC31-6808

MVS/DFP 3.2 System Programming Reference, SC26-4567
IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version
1.4.2 Diagnostics Guide, SC34-6358

Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully.

You can perform most tasks required to set up, run, and maintain your CICS system in one of these ways:

- using a 3270 emulator logged on to CICS
- using a 3270 emulator logged on to TSO
- using a 3270 emulator as an MVS system console

IBM Personal Communications provides 3270 emulation with accessibility features for people with disabilities. You can use this product to provide the accessibility features you need in your CICS system.

Index

A

- absolute expression 13
- access to system information
 - INQUIRE STORAGE command 433
- ACCESSMETHOD option
 - INQUIRE CONNECTION command 260
 - INQUIRE DSNNAME command 307
 - INQUIRE FILE command 335
 - INQUIRE TERMINAL command 485
- ACQSTATUS option
 - INQUIRE CONNECTION command 260
 - INQUIRE TERMINAL command 485
 - SET CONNECTION command 615
 - SET MODENAME command 692
 - SET TERMINAL command 735
- ACQUIRE TERMINAL command 31
 - conditions 33
- ACTION option
 - SET DSNNAME command 649
 - SET SYSDUMPCODE command 713
 - SET TRANDUMPCODE command 753
 - SET UOWLINK command 762
- ACTIVE option
 - INQUIRE MODENAME command 378
 - INQUIRE TRANCLASS command 512
- ACTIVITY option
 - INQUIRE TASK command 454
- ACTIVITYID option
 - INQUIRE TASK command 454
- ACTJVMTCBS option
 - INQUIRE DISPATCHER command 296
- ACTOPENTCBS option
 - INQUIRE DISPATCHER command 296
 - INQUIRE SYSTEM command 443
- ACTSOCKETS option
 - INQUIRE TCPIP command 466
- ACTTHRDTCBS option
 - INQUIRE DISPATCHER command 296
- ACTXPTCBS option
 - INQUIRE DISPATCHER command 297
- ADAPTERTYPE option
 - INQUIRE EPADAPTER command 318
- ADD command
 - CSD 124
- ADD option
 - INQUIRE FILE command 335
 - SET FILE command 662
- ADDRESS option
 - INQUIRE STORAGE command 433
- AFFINITY option
 - SET CONNECTION command 615
- AFTER option
 - INQUIRE REQID command 423
- AGE option
 - INQUIRE JVM command 365
 - INQUIRE UOW command 536
- AIBRIDGE option
 - INQUIRE AUTOINSTALL command 230
 - SET AUTOINSTALL command 609
- AKP option
 - INQUIRE SYSTEM command 443
 - SET SYSTEM command 717
- ALIGNED attribute
 - PL/I 12
- ALL option
 - PERFORM STATISTICS command 600
- ALLOCAGE option
 - INQUIRE JVM command 366
- ALTER command
 - CSD 126
- ALTPAGEHT option
 - INQUIRE TERMINAL command 485
- ALTPAGEWD option
 - INQUIRE TERMINAL command 485
- ALTPRINTER option
 - INQUIRE TERMINAL command 485
 - SET TERMINAL command 735
- ALTPRTCOPYST option
 - INQUIRE TERMINAL command 485
 - SET TERMINAL command 735
- ALTSCRNHT option
 - INQUIRE TERMINAL command 485
- ALTSCRNWD option
 - INQUIRE TERMINAL command 485
- ALTSUFFIX option
 - INQUIRE TERMINAL command 485
- ANALYZERSTAT option
 - INQUIRE URIMAP command 555
- APIST option
 - INQUIRE EXITPROGRAM command 327
 - INQUIRE PROGRAM command 410
- APLKYBDST option
 - INQUIRE TERMINAL command 485
- APLTEXTST option
 - INQUIRE TERMINAL command 485
- APPEND command
 - CSD 129
- APPENDCRLF option
 - INQUIRE DOCTEMPLATE command 303
- APPLDATA option
 - INQUIRE ASSOCIATION command 212
- APPLID option
 - INQUIRE ASSOCIATION command 212
 - INQUIRE IPCONN command 348
- APPLNAMEST option
 - INQUIRE MONITOR command 381
- ARCHIVEFILE option
 - INQUIRE WEBSERVICE command 567
- argument lengths 14
- argument values
 - assembler language 13
 - C 11
 - COBOL 11
 - PL/I 12
- ASCII option
 - INQUIRE TERMINAL command 485
- assembler language
 - argument values 13
- ASSERTED option
 - INQUIRE CORBASERVER command 272
- association data 212, 222
- ASSOCIATION LIST, INQUIRE command 222
- ASSOCIATION option
 - INQUIRE ASSOCIATION command 212
- ASSOCIATION, INQUIRE command 212
- AT option
 - INQUIRE REQID command 423
- ATIFACILITY option
 - INQUIRE TDQUEUE command 477
 - SET TDQUEUE command 730
- ATISTATUS option
 - INQUIRE TERMINAL command 485
 - SET TERMINAL command 736
- ATITERMID option
 - INQUIRE TDQUEUE command 478
 - SET TDQUEUE command 730
- ATITRANID option
 - INQUIRE TDQUEUE command 478
 - SET TDQUEUE command 730
- ATIUSERID option
 - INQUIRE TDQUEUE command 478
 - SET TDQUEUE command 730
- ATOMSERVICE CVDA value
 - EXTRACT STATISTICS command 209
- ATOMSERVICE option
 - CREATE ATOMSERVICE command 43
 - DISCARD ATOMSERVICE command 170
 - INQUIRE ATOMSERVICE command 226
 - INQUIRE URIMAP command 555
 - PERFORM STATISTICS command 600
- ATOMSERVICE, CREATE command 42
- ATOMTYPE option
 - INQUIRE ATOMSERVICE command 226
- ATTACHSEC option
 - INQUIRE TCPIPSERVICE command 468

ATTACHTIME option
 INQUIRE TASK command 454

ATTRIBUTES option
 CREATE ATOMSERVICE
 command 43
 CREATE BUNDLE command 44, 72
 CREATE CONNECTION
 command 47
 CREATE CORBASERVER
 command 50
 CREATE DB2CONN command 51
 CREATE DB2ENTRY command 55
 CREATE DB2TRAN command 56
 CREATE DJAR command 58
 CREATE DOCTEMPLATE
 command 61
 CREATE ENQMODEL command 62
 CREATE FILE command 63
 CREATE IPCONN command 69
 CREATE JOURNALMODEL
 command 70
 CREATE LIBRARY command 75
 CREATE LSRPOOL command 78
 CREATE MAPSET command 80
 CREATE MQCONN command 82
 CREATE PARTITIONSET
 command 84
 CREATE PARTNER command 86
 CREATE PIPELINE command 88
 CREATE PROCESSTYPE
 command 89
 CREATE PROFILE command 92
 CREATE PROGRAM command 93
 CREATE REQUESTMODEL
 command 96
 CREATE SESSIONS command 99
 CREATE TCPIP SERVICE
 command 102
 CREATE TDQUEUE command 105
 CREATE TERMINAL command 108
 CREATE TRANCLASS
 command 110
 CREATE TRANSACTION
 command 113
 CREATE TSMODEL command 115
 CREATE TYPETERM command 119
 CREATE URIMAP command 122
 CREATE WEBSERVICE
 command 123

ATTRLEN option
 CREATE ATOMSERVICE
 command 43
 CREATE BUNDLE command 45, 72
 CREATE CONNECTION
 command 48
 CREATE CORBASERVER
 command 50
 CREATE DJAR command 59
 CREATE DOCTEMPLATE
 command 61
 CREATE ENQMODEL command 63
 CREATE FILE command 63, 75
 CREATE IPCONN command 69
 CREATE JOURNALMODEL
 command 71
 CREATE LSRPOOL command 79
 CREATE MAPSET command 81

ATTRLEN option (*continued*)
 CREATE PARTITIONSET
 command 84
 CREATE PARTNER command 86
 CREATE PIPELINE command 88
 CREATE PROCESSTYPE
 command 89
 CREATE PROFILE command 92
 CREATE PROGRAM command 93
 CREATE REQUESTMODEL
 command 96
 CREATE SESSIONS command 99
 CREATE TCPIP SERVICE
 command 102
 CREATE TDQUEUE command 105
 CREATE TERMINAL command 108
 CREATE TRANCLASS
 command 110
 CREATE TRANSACTION
 command 113
 CREATE TSMODEL command 115
 CREATE TYPETERM command 119
 CREATE URIMAP command 122
 CREATE WEBSERVICE
 command 123

AUDALARMST option
 INQUIRE TERMINAL command 485

AUDITLEVEL option
 INQUIRE PROCESSTYPE
 command 405, 703

AUDITLOG option
 INQUIRE PROCESSTYPE
 command 406

AUTHENTICATE option
 INQUIRE TCPIP SERVICE
 command 468
 INQUIRE URIMAP command 555

AUTHORITY option
 INQUIRE EPADAPTER
 command 318

authorization failures 16

AUTHUSERID option
 INQUIRE EPADAPTER
 command 319

AUTINSTMODEL option
 DISCARD AUTINSTMODEL
 command 171
 INQUIRE AUTINSTMODEL
 command 229

AUTINSTMODEL, DISCARD
 command 171

AUTINSTMODEL, INQUIRE
 command 229

AUTOCONNECT option
 INQUIRE CONNECTION
 command 260
 INQUIRE IPCONN command 348
 INQUIRE MODENAME
 command 378
 INQUIRE TERMINAL command 485

AUTOINSTALL option
 COLLECT STATISTICS command 33
 PERFORM STATISTICS
 command 600

AUTOINSTALL, INQUIRE
 command 230

AUTOINSTALL, SET command 609

automatic initiate descriptor 485

automatic installation of terminals 230

AUTOPUBLISH option
 INQUIRE CORBASERVER
 command 272

AUTOSTARTST option
 INQUIRE CLASSCACHE
 command 256
 PERFORM CLASSCACHE
 command 580
 SET CLASSCACHE command 613

AUXSTATUS option
 INQUIRE TRACEDEST
 command 505
 SET TRACEDEST command 743

AVAILABILITY option
 INQUIRE DSNNAME 307
 SET DSNNAME 650

AVAILABLE option
 INQUIRE MODENAME
 command 378
 SET MODENAME command 692

B

BACKLOG option
 INQUIRE TCPIP SERVICE
 command 469
 SET TCPIP SERVICE command 726

BACKTRANSST option
 INQUIRE TERMINAL command 485

BACKUPTYPE option
 INQUIRE DSNNAME command 307

BASEDSNAME option
 INQUIRE DSNNAME command 308
 INQUIRE FILE command 335

BASESCOPE option
 INQUIRE BUNDLE command 237

batch backout utility 306

BEAN option
 COLLECT STATISTICS command 33
 INQUIRE BEAN command 232
 PERFORM STATISTICS
 command 600

BEAN, INQUIRE command 232

BEANNAME option
 INQUIRE REQUESTMODEL
 command 427

BINDFILE option
 INQUIRE ATOMSERVICE
 command 226

BINDING option
 INQUIRE WEBSERVICE
 command 567

BLOCKFORMAT option
 INQUIRE FILE command 336
 INQUIRE TDQUEUE command 478

BLOCKKEYLEN option
 INQUIRE FILE command 336

BLOCKSIZE option
 INQUIRE FILE command 336
 INQUIRE TDQUEUE command 478

BREXIT option
 INQUIRE TRANSACTION
 command 519

- BRFACILITY option
 - INQUIRE BRFACILITY command 233
 - INQUIRE TASK command 454
 - BRFACILITY, INQUIRE command 233
 - BRFACILITY, SET command 610
 - BRIDGE option
 - INQUIRE TASK command 455
 - BROWSE option
 - INQUIRE FILE command 336
 - SET FILE command 662
 - browsing
 - AUTINSTMODEL entries 229
 - BEAN entries 232
 - BRFACILITY entries 233
 - bundles 236, 241, 244, 247, 249, 318, 322
 - CFDTPPOOL entries 255
 - CONNECTION entries 259
 - CORBASERVER entries 271
 - DB2ENTRY entries 286
 - DB2TRAN entries 292
 - DJAR entries 299
 - DOCTEMPLATE entries 302
 - FILE entries 334, 363, 374, 435
 - inside bundles 239
 - IPCONN entries 348
 - JOURNALMODEL entries 360
 - JVM entries 365
 - JVM profiles 369
 - MODENAME entries 378
 - NETNAME entries 393
 - OSGIBUNDLE 394
 - OSGISERVICE 397
 - PARTNER entries 399
 - PIPELINES 401
 - PROCESSTYPE entries 405
 - PROFILE entries 408
 - PROGRAM entries 410
 - REQUESTMODEL entries 426
 - TDQUEUE entries 477
 - TERMINAL entries 485
 - TRANCLASS entries 512
 - TRANDUMPCODE entries 515
 - TRANSACTION entries 519
 - Workrequests 572
 - browsing resource definitions 20
 - browsing rules 22
 - BUNDLE CVDA value
 - EXTRACT STATISTICS command 209
 - BUNDLE option
 - CREATE BUNDLE command 45
 - INQUIRE BUNDLE command 237
 - INQUIRE BUNDLEPART command 239
 - PERFORM STATISTICS command 600
 - BUNDLE, CREATE command 44
 - BUNDLE, DISCARD command 172
 - BUNDLE, INQUIRE command 235
 - BUNDLE, SET command 611
 - BUNDLEDIR option
 - INQUIRE BUNDLE command 237
 - BUNDLEPART option
 - INQUIRE BUNDLEPART command 239
 - BUNDLEPART, INQUIRE command 239
 - BUSY option
 - SET FILE command 662
- C**
- C language
 - argument values 11
 - CACHEFREE option
 - INQUIRE CLASSCACHE command 257
 - CACHESIZE option
 - INQUIRE CLASSCACHE command 257
 - INQUIRE DOCTEMPLATE command 303
 - PERFORM CLASSCACHE command 581
 - caching of document templates 646
 - CALLER option
 - PERFORM DUMP command 587
 - CALLERLENGTH option
 - PERFORM DUMP command 587
 - CANCEL option
 - SET CONNECTION command 617
 - SET IPCONN command 678
 - CAPDATAPRED, INQUIRE command 241
 - CAPINFOSRCE, INQUIRE command 244
 - CAPOPTPRED, INQUIRE command 246
 - CAPTUREPOINT option
 - INQUIRE CAPTURESPEC command 249
 - CAPTUREPTYPE option
 - INQUIRE CAPTURESPEC command 249
 - CAPTURESPEC CVDA value
 - EXTRACT STATISTICS command 210
 - CAPTURESPEC option
 - INQUIRE CAPTURESPEC command 250
 - PERFORM STATISTICS command 600
 - CAPTURESPEC, INQUIRE command 249
 - CCSID option
 - INQUIRE WEBSERVICE command 567
 - CDSASIZE option
 - INQUIRE SYSTEM command 443
 - CECI transaction 1
 - CEDF transaction 1
 - CEDFSTATUS option
 - INQUIRE PROGRAM command 410
 - SET PROGRAM command 705
 - CEMT transaction
 - function provided by INQUIRE and SET commands 1
 - CERTIFICATE option
 - INQUIRE CORBASERVER command 272
 - INQUIRE IPCONN command 348
 - INQUIRE TCPIP SERVICE command 469
 - INQUIRE URIMAP command 555
 - CETR transaction
 - function provided by INQUIRE and SET commands 1
 - CFDTPPOOL option
 - INQUIRE CFDTPPOOL command 255
 - INQUIRE FILE command 336
 - SET FILE command 662
 - CFDTPPOOL, INQUIRE command 254
 - char-expr argument, CICS command format 6
 - CHARACTERSET option
 - INQUIRE URIMAP command 556
 - CICS value
 - INQUIRE TCPIP command 466
 - CICS-supplied security 2
 - CICS-value data area (CVDA) 8
 - CICSSTATUS option
 - INQUIRE SYSTEM command 443
 - CICSSYS option
 - INQUIRE SYSTEM command 443
 - CICSTSLEVEL option
 - INQUIRE SYSTEM command 443
 - CIPHERS option
 - INQUIRE CORBASERVER command 273
 - INQUIRE IPCONN command 348
 - INQUIRE TCPIP SERVICE command 470
 - INQUIRE URIMAP command 556
 - PARTNER option
 - PARTNER command 348
 - CLASSCACHE, INQUIRE command 256
 - CLASSCACHE, PERFORM command 580
 - CLASSCACHE, SET command 612
 - CLASSCACHES option
 - INQUIRE JVM command 366
 - INQUIRE JVMPROFILE command 369
 - CLIENTAUTH value
 - INQUIRE TCPIP SERVICE command 474
 - CLIENTCERT option
 - INQUIRE CORBASERVER command 273
 - CLIENTIPADDR option
 - INQUIRE ASSOCIATION command 212
 - INQUIRE WORKREQUEST command 572
 - CLIENTLOC option
 - CLIENTLOC command 212, 348
 - CLIENTPORT option
 - INQUIRE ASSOCIATION command 212
 - CLNTIP6ADDR option
 - INQUIRE WORKREQUEST command 572
 - CLNTIPFAMILY option
 - INQUIRE ASSOCIATION command 212
 - INQUIRE WORKREQUEST command 572
 - CLOSED value
 - INQUIRE TCPIP command 466
 - INQUIRE TCPIP SERVICE command 473

CLOSED value (*continued*)
 SET TCPIP command 725
 SET TCPIPSERVICE command 727
 CLOSETIMEOUT option
 INQUIRE TCPIPSERVICE
 command 470
 CLOSING value
 INQUIRE TCPIP command 466
 INQUIRE TCPIPSERVICE
 command 473
 CMDPROTECT option
 INQUIRE SYSTEM command 443
 CMDSEC option
 INQUIRE TASK command 455
 INQUIRE TRANSACTION
 command 520
 COBOL
 argument values 11
 COBOLTYPE option
 INQUIRE PROGRAM command 410
 COLDSTATUS option
 INQUIRE SYSTEM command 444
 COLLECT STATISTICS
 conditions 33
 COLLECT STATISTICS command 33
 COLORST option
 INQUIRE TERMINAL command 485
 command interpreter transaction
 (CECI) 1
 command security checking 16
 command, CREATE FILE 63
 command, CREATE LIBRARY 73
 commands
 format, arguments 2
 COMPID option
 INQUIRE TRACETYPE
 command 510
 SET TRACETYPE command 749
 COMPLETE option
 CREATE CONNECTION
 command 48
 CREATE TERMINAL command 108
 COMPRESST option
 INQUIRE MONITOR command 382
 SET MONITOR command 695
 CONCURRENCY option
 INQUIRE PROGRAM command 410
 CONCURRENTST option
 INQUIRE EXITPROGRAM
 command 328
 conditions
 ACQUIRE TERMINAL command 33
 COLLECT STATISTICS command 33
 CREATE ATOMSERVICE
 command 43
 CREATE BUNDLE command 45
 CREATE CONNECTION
 command 48
 CREATE CORBASERVER
 command 50
 CREATE DB2ENTRY command 55
 CREATE DB2TRAN command 57
 CREATE DJAR command 59
 CREATE DOCTEMPLATE
 command 61
 CREATE ENQMODEL command 63
 CREATE FILE command 63, 76

conditions (*continued*)
 CREATE IPCONN command 69
 CREATE JOURNALMODEL
 command 71
 CREATE JVMSERVER command 73
 CREATE LSRPOOL command 79
 CREATE MAPSET command 81
 CREATE MQCONN command 82
 CREATE PARTITIONSET
 command 84
 CREATE PARTNER command 86
 CREATE PIPELINE command 88
 CREATE PROCESSTYPE
 command 90
 CREATE PROFILE command 92
 CREATE PROGRAM command 93
 CREATE REQUESTMODEL
 command 97
 CREATE SESSIONS command 99
 CREATE TCPIPSERVICE
 command 102
 CREATE TDQUEUE command 106
 CREATE TERMINAL command 109
 CREATE TRANCLASS
 command 110
 CREATE TRANSACTION
 command 113
 CREATE TSMODEL command 115
 CREATE TYPETERM command 119
 CREATE URIMAP command 122
 CREATE WEBSERVICE
 command 124
 CSD ADD command 125
 CSD ALTER command 128
 CSD APPEND command 130
 CSD COPY command 132
 CSD DEFINE command 135
 CSD DELETE command 138
 CSD DISCONNECT command 139
 CSD ENDBRGROUP command 140
 CSD ENDBRLIST command 141
 CSD ENDBRRSRCE command 141
 CSD GETNEXTGROUP
 command 142
 CSD GETNEXTLIST command 143
 CSD GETNEXTSRCE
 command 145
 CSD INQUIREGROUP
 command 146
 CSD INQUIRELIST command 147
 CSD INQUIRERSRCE command 149
 CSD INSTALL command 152
 CSD LOCK command 154
 CSD REMOVE command 155
 CSD RENAME command 158
 CSD STARTBRGROUP
 command 159
 CSD STARTBRLIST command 161
 CSD STARTBRRSRCE command 162
 CSD UNLOCK command 163
 CSD USERDEFINE command 165
 DB2CONN command 51
 DISABLE PROGRAM command 169
 DISCARD ATOMSERVICE
 command 170
 DISCARD AUTINSTMODEL
 command 171

conditions (*continued*)
 DISCARD BUNDLE command 172
 DISCARD CONNECTION
 command 173
 DISCARD CORBASERVER
 command 174
 DISCARD DB2ENTRY command 176
 DISCARD DJAR command 178
 DISCARD DOCTEMPLATE
 command 178
 DISCARD ENQMODEL
 command 179
 DISCARD FILE command 180
 DISCARD IPCONN command 181
 DISCARD JOURNALMODEL
 command 182
 DISCARD JOURNALNAME
 command 183
 DISCARD JVMSERVER
 command 184
 DISCARD LIBRARY command 184
 DISCARD PARTNER command 186
 DISCARD PIPELINE command 187
 DISCARD PROCESSTYPE
 command 188
 DISCARD PROFILE command 189
 DISCARD PROGRAM command 190
 DISCARD REQUESTMODEL
 command 191
 DISCARD TCPIPSERVICE
 command 191
 DISCARD TDQUEUE command 192
 DISCARD TERMINAL
 command 194
 DISCARD TRANCLASS
 command 195
 DISCARD TRANSACTION
 command 196
 DISCARD TSMODEL command 197
 DISCARD URIMAP command 197
 DISCARD WEBSERVICE
 command 198
 ENABLE PROGRAM command 203
 EXTRACT EXIT command 206
 EXTRACT STATISTICS
 command 211
 INQUIRE ASSOCIATION
 command 212
 INQUIRE ASSOCIATION LIST
 command 222
 INQUIRE ATOMSERVICE
 command 228
 INQUIRE AUTINSTMODEL
 command 229
 INQUIRE AUTOINSTALL
 command 231
 INQUIRE BEAN command 232
 INQUIRE BRFACTILITY
 command 235
 INQUIRE BUNDLE command 238
 INQUIRE BUNDLEPART
 command 239
 INQUIRE CAPDATAPRED
 command 241
 INQUIRE CAPINFORSCE
 command 246

conditions (continued)

INQUIRE CAPOPTPRED
 command 248
 INQUIRE CAPTURESPEC
 command 254
 INQUIRE CFDTPOOL command 255
 INQUIRE CLASSCACHE
 command 258
 INQUIRE command 23
 INQUIRE CONNECTION
 command 267
 INQUIRE CORBASERVER
 command 275, 301
 INQUIRE DB2CONN command 276
 INQUIRE DB2ENTRY command 291
 INQUIRE DB2TRAN command 293
 INQUIRE DELETSHIPED 295
 INQUIRE DISPATCHER
 command 298
 INQUIRE DOCTEMPLATE
 command 305
 INQUIRE DSNNAME command 311
 INQUIRE DUMPDS command 313
 INQUIRE ENQMODEL 317
 INQUIRE EPADAPTER
 command 321
 INQUIRE EVENTBINDING
 command 324
 INQUIRE EVENTPROCESS
 command 325
 INQUIRE EXCI command 326
 INQUIRE EXITPROGRAM
 command 332
 INQUIRE FILE command 346
 INQUIRE HOST command 348
 INQUIRE IPCONN command 348
 INQUIRE IPFACILITY command 358
 INQUIRE IRC command 359
 INQUIRE JOURNALMODEL
 command 362
 INQUIRE JOURNALNAME
 command 364
 INQUIRE JVM command 366, 369
 INQUIRE JVMPOOL command 368
 INQUIRE JVMSERVER
 command 373
 INQUIRE LIBRARY command 378
 INQUIRE MODENAME
 command 378, 436
 INQUIRE MONITOR command 384
 INQUIRE MQCONN command 384
 INQUIRE MQINI command 390
 INQUIRE MVSTCB command 392
 INQUIRE OSGIBUNDLE
 command 396
 INQUIRE OSGISERVICE
 command 398
 INQUIRE PARTNER command 399
 INQUIRE PIPELINE command 404
 INQUIRE PROCESSTYPE
 command 407, 704
 INQUIRE PROFILE command 409
 INQUIRE PROGRAM command 410
 INQUIRE REQID command 425
 INQUIRE REQUESTMODEL
 command 428
 INQUIRE rrms command 429

conditions (continued)

INQUIRE STATISTICS command 432
 INQUIRE STORAGE command 434
 INQUIRE SUBPOOL 437
 INQUIRE SYSDUMPCODE
 command 440
 INQUIRE SYSTEM command 452
 INQUIRE TASK command 463
 INQUIRE TASK LIST command 463
 INQUIRE TCLASS command 465
 INQUIRE TCPIP command 467
 INQUIRE TCPIPSERVICE
 command 474
 INQUIRE TDQUEUE command 484
 INQUIRE TEMPSTORAGE 485
 INQUIRE TERMINAL command 485
 INQUIRE TRACEDEST
 command 507
 INQUIRE TRACEFLAG
 command 509
 INQUIRE TRACETYPE
 command 511
 INQUIRE TRANCLASS
 command 514
 INQUIRE TRANDUMPCODE
 command 517
 INQUIRE TRANSACTION
 command 526
 INQUIRE TSMODEL 529
 INQUIRE TSPOOL 531
 INQUIRE TSQNAME 534
 INQUIRE TSQUEUE 534
 INQUIRE UOW command 539
 INQUIRE UOWDSNFAL
 command 544
 INQUIRE UOWENQ command 549
 INQUIRE UOWLINK command 553
 INQUIRE URIMAP command 561
 INQUIRE VOLUME command 561
 INQUIRE VTAM command 564
 INQUIRE WEB command 565
 INQUIRE WEBSERVICE
 command 571
 INQUIRE WORKREQUEST 575
 PERFORM CLASSCACHE
 command 582
 PERFORM CORBASERVER
 command 584
 PERFORM DJAR command 587
 PERFORM DUMP command 587
 PERFORM ENDAFFINITY
 command 590
 PERFORM JVMPOOL command 592
 PERFORM PIPELINE command 594
 PERFORM RESETTIME
 command 594
 PERFORM SECURITY REBUILD
 command 595
 PERFORM SHUTDOWN
 command 598
 PERFORM STATISTICS RECORD
 command 603
 RESYNC ENTRYNAME
 command 608
 SET ATOMSERVICE command 608
 SET AUTOINSTALL command 610
 SET BRFACILITY 611

conditions (continued)

SET BUNDLE command 612
 SET CLASSCACHE command 613
 SET CONNECTION command 620
 SET CORBASERVER command 624
 SET DB2CONN command 624
 SET DB2ENTRY command 639
 SET DB2TRAN command 640
 SET DELETSHIPED command 643
 SET DISPATCHER command 645
 SET DOCTEMPLATE command 647
 SET DSNNAME command 652
 SET DUMPDS command 658
 SET ENQMODEL command 659
 SET EPADAPTER command 660
 SET EVENTBINDING command 661
 SET EVENTPROCESS command 662
 SET FILE command 662
 SET HOST command 676
 SET IPCONN command 680
 SET IRC command 682
 SET JOURNALNAME command 684
 SET JVMPOOL command 686
 SET JVMSERVER command 688
 SET MODENAME command 692
 SET MONITOR command 697
 SET MQCONN command 698
 SET NETNAME command 701
 SET PIPELINE command 702
 SET PROGRAM command 690, 708
 SET STATISTICS command 711
 SET SYSDUMPCODE command 715
 SET SYSTEM command 720
 SET TASK command 723
 SET TCLASS command 724
 SET TCPIP command 725
 SET TCPIPSERVICE command 727
 SET TDQUEUE command 731
 SET TEMPSTORAGE command 733
 SET TERMINAL command 741
 SET TRACEDEST command 745
 SET TRACEFLAG command 748
 SET TRACETYPE command 750
 SET TRANCLASS command 752
 SET TRANDUMPCODE
 command 755
 SET TRANSACTION command 759
 SET TSQNAME command 760
 SET TSQUEUE command 760
 SET UOW command 761
 SET UOWLINK command 762
 SET URIMAP command 764
 SET VOLUME command 765
 SET VTAM command 767
 SET WEBSERVICE command 770
 SET WORKREQUEST command 771
 XMLTRANSFORM command 579,
 773
 CONFIGDATA1 option
 INQUIRE EPADAPTER
 command 319
 CONFIGFILE option
 INQUIRE ATOMSERVICE
 command 227
 CONNECTION
 SET CONNECTION command 615

CONNECTION option
 COLLECT STATISTICS command 33
 CREATE CONNECTION command 48
 INQUIRE CONNECTION command 261
 INQUIRE MODENAME command 378
 PERFORM STATISTICS command 600
 SET MODENAME command 692
CONNECTION, CREATE command 45
CONNECTION, DISCARD command 172
CONNECTION, INQUIRE command 258
CONNECTION, SET command 614, 762
CONNECTIONS option
 INQUIRE TCPIP SERVICE command 470
CONNECTST option
 INQUIRE EXITPROGRAM command 328
CONNSTATUS option
 INQUIRE CFDTPOOL command 255
 INQUIRE CONNECTION command 261
 INQUIRE IPCONN command 348
 INQUIRE TSPool command 530
 SET CONNECTION command 615
 SET IPCONN command 677
CONNTYPE option
 INQUIRE CONNECTION command 262
CONSOLE option
 INQUIRE TERMINAL command 485
CONSOLES option
 INQUIRE AUTOINSTALL command 230
 SET AUTOINSTALL command 609
CONTAINER option
 INQUIRE CAPDATAPRED command 241
 INQUIRE CAPINFOSRCE command 245
 INQUIRE WEBSERVICE command 567
CONVERSEST option
 INQUIRE MONITOR command 382
 SET MONITOR command 695
CONVERTER option
 INQUIRE URIMAP command 556
COPID
 option of DSNCRCT macro 624
COPY command
 CSD 131
COPY option
 INQUIRE PROGRAM command 410
 SET DOCTEMPLATE command 646
 SET PROGRAM command 705
COPYST option
 INQUIRE TERMINAL command 485
CORBASERVER option
 COLLECT STATISTICS command 33
 CREATE CORBASERVER command 50
 INQUIRE BEAN command 232
CORBASERVER option (continued)
 INQUIRE CORBASERVER command 273
 INQUIRE REQUESTMODEL command 427
 INQUIRE WORKREQUEST command 573
 PERFORM CORBASERVER command 583
 PERFORM STATISTICS command 600
CORBASERVER, CREATE command 49
CORBASERVER, DISCARD command 174
CORBASERVER, INQUIRE command 271
CORBASERVER, PERFORM 582
CORBASERVER, SET command 622
CORBSERVER option
 INQUIRE DJAR command 300
 PERFORM DJAR command 586
 SET CORBASERVER command 623
CORRELID option
 INQUIRE TERMINAL command 485
CREATE ATOMSERVICE command 42
 conditions 43
CREATE BUNDLE command 44
 conditions 45
CREATE CONNECTION command 45
 conditions 48
CREATE CORBASERVER command 49
 conditions 50
CREATE DB2CONN command 51
CREATE DB2ENTRY command 54
 conditions 55
CREATE DB2TRAN command 56
 conditions 57
CREATE DJAR command 58
 conditions 59
CREATE DOCTEMPLATE command 60
 conditions 61
CREATE ENQMODEL command 62
 conditions 63
CREATE FILE command 63
 conditions 63, 76
CREATE IPCONN command 66
 conditions 69
CREATE JOURNALMODEL command 70
 conditions 71
CREATE JVMSEVER command 72
 conditions 73
CREATE LIBRARY command 73
CREATE LSRPOOL command 77
 conditions 79
CREATE MAPSET command 80
 conditions 81
CREATE MQCONN command 82
 conditions 82
CREATE PARTITIONSET command 83
 conditions 84
CREATE PARTNER command 85
 conditions 86
CREATE PIPELINE command 87
 conditions 88
CREATE PROCESSTYPE command 89
 conditions 90
CREATE PROFILE command 90
 conditions 92
CREATE PROGRAM command 93
 conditions 93
CREATE REQUESTMODEL command 95
 conditions 97
CREATE SESSIONS command 97
 conditions 99
CREATE TCPIP SERVICE command 100
 conditions 102
CREATE TDQUEUE command 103
 conditions 106
CREATE TERMINAL command 106
 conditions 109
CREATE TRANCLASS command 109
 conditions 110
CREATE TRANSACTION command 111
 conditions 113
CREATE TSMODEL command 114
 conditions 115
CREATE TYPETERM command 116
 conditions 119
CREATE URIMAP command 120
 conditions 122
CREATE WEBSERVICE command 123
 conditions 124
CREATESESS option
 INQUIRE TERMINAL command 485
 SET TERMINAL command 736
creating resource definitions 24
CRITICAL value 471
CRITICALST option
 INQUIRE LIBRARY command 375
 SET LIBRARY command 689
CRLPROFILE option
 INQUIRE TCPIP command 466
CSD
 ADD command 124
 ALTER command 126
 APPEND command 129
 COPY command 131
 DEFINE command 133
 DELETE command 137
 DISCONNECT command 139
 ENDBRGROUP command 139
 ENDBRLIST command 140
 ENDBRRSRCE command 141
 GETNEXTGROUP command 142
 GETNEXTLIST command 143
 GETNEXTSRCE command 143
 INQUIREGROUP command 146
 INQUIRELIST command 147
 INQUIERSRCE command 147
 INSTALL command 150
 LOCK command 153
 REMOVE command 154
 RENAME command 156
 STARTBRGROUP command 159
 STARTBRLIST command 160
 STARTBRRSRCE command 161
 UNLOCK command 162
 USERDEFINE command 163
CSD ADD command
 conditions 125
CSD ALTER command
 conditions 128

CSD APPEND command
 conditions 130

CSD COPY command
 conditions 132

CSD DEFINE command
 conditions 135

CSD DELETE command
 conditions 138

CSD DISCONNECT command
 conditions 139

CSD ENDBRGROUP command
 conditions 140

CSD ENDBRLIST command
 conditions 141

CSD ENDBRRSRCE command
 conditions 141

CSD GETNEXTGROUP command
 conditions 142

CSD GETNEXTLIST command
 conditions 143

CSD GETNEXTRSRCE command
 conditions 145

CSD INQUIREGROUP command
 conditions 146

CSD INQUIRELIST command
 conditions 147

CSD INQUIRERSRCE command
 conditions 149

CSD INSTALL command
 conditions 152

CSD LOCK command
 conditions 154

CSD REMOVE command
 conditions 155

CSD RENAME command
 conditions 158

CSD STARTBRGROUP command
 conditions 159

CSD STARTBRLIST command
 conditions 161

CSD STARTBRRSRCE command
 conditions 162

CSD UNLOCK command
 conditions 163

CSD USERDEFINE command
 conditions 165

CTERM option
 DSNCRCT macro 624

CTX option of DSNCRCT macro 624

CURAUXDS option
 INQUIRE TRACEDEST
 command 506

CURRENT option
 INQUIRE SYSDUMPCODE
 command 438
 INQUIRE TCLASS command 465
 INQUIRE TRANDUMPCODE
 command 515

CURRENTDDS option
 INQUIRE DUMPDS command 313

CURRENTPROG option 455

CURREQS option
 INQUIRE AUTOINSTALL
 command 231

CURRPGM option
 INQUIRE CAPTURESPEC
 command 250

CURRPGMOP option
 INQUIRE CAPTURESPEC
 command 250

CURRTRANID option
 INQUIRE CAPTURESPEC
 command 251

CURRTRANIDOP option
 INQUIRE CAPTURESPEC
 command 251

CURRUSERID option
 INQUIRE CAPTURESPEC
 command 251

CURRUSERIDOP option
 INQUIRE CAPTURESPEC
 command 251

CVDA (CICS-value data area) 8
 argument values 6
 command format 6
 example code 9
 listed in numerical and alphabetical
 order 775

CVDA values
 ALLVALUES
 INQUIRE CAPTURESPEC
 command 250, 251, 253

ASYNCHRONOUS
 INQUIRE EPADAPTER
 command 320

ATOM
 INQUIRE URIMAP
 command 560

BASIC
 INQUIRE URIMAP
 command 555

CATEGORY
 INQUIRE ATOMSERVICE
 command 226

CBE
 INQUIRE EPADAPTER
 command 320

CBER
 INQUIRE EPADAPTER
 command 320

CCE
 INQUIRE EPADAPTER
 command 320

CFE
 INQUIRE EPADAPTER
 command 320

CLIENT
 INQUIRE URIMAP
 command 560

COLLECTION
 INQUIRE ATOMSERVICE
 command 226

CONTAINER
 INQUIRE CAPTURESPEC
 command 253

CONTEXT
 INQUIRE EPADAPTER
 command 318

CURRENTPGM
 INQUIRE CAPTURESPEC
 command 253

CUSTOM
 INQUIRE EPADAPTER
 command 318, 319

CVDA values (*continued*)
 DISABLED
 INQUIRE ATOMSERVICE
 command 227
 INQUIRE EPADAPTER
 command 320
 INQUIRE EVENTBINDING
 command 323
 INQUIRE HOST command 347
 INQUIRE URIMAP
 command 557, 559
 SET ATOMSERVICE
 command 608
 SET EPADAPTER command 660
 SET EVENTBINDING
 command 661
 SET HOST command 676
 SET URIMAP command 764

DISABLEDHOST
 INQUIRE URIMAP
 command 557

DOESNOTEQUAL
 INQUIRE CAPDATAPRED
 command 242
 INQUIRE CAPOPTPRED
 command 247
 INQUIRE CAPTURESPEC
 command 250, 251, 252, 253

DOESNOTEXIST
 INQUIRE CAPDATAPRED
 command 242
 INQUIRE CAPOPTPRED
 command 247

DOESNOTSTART
 INQUIRE CAPDATAPRED
 command 242
 INQUIRE CAPOPTPRED
 command 247
 INQUIRE CAPTURESPEC
 command 250, 251, 252, 253

ENABLED
 INQUIRE ATOMSERVICE
 command 227
 INQUIRE EPADAPTER
 command 320
 INQUIRE EVENTBINDING
 command 323
 INQUIRE HOST command 347
 INQUIRE URIMAP
 command 556, 559
 SET ATOMSERVICE
 command 608
 SET EPADAPTER command 660
 SET EVENTBINDING
 command 661
 SET HOST command 676
 SET URIMAP command 764

EQUALS
 INQUIRE CAPDATAPRED
 command 242
 INQUIRE CAPOPTPRED
 command 247
 INQUIRE CAPTURESPEC
 command 251, 252, 253

EVENT
 INQUIRE CAPTURESPEC
 command 253

CVDA values (continued)

FEED
 INQUIRE ATOMSERVICE
 command 226

FILE
 INQUIRE ATOMSERVICE
 command 228
 INQUIRE CAPTURESPEC
 command 253

GOHIGHERTHAN
 INQUIRE CAPOPTPRED
 command 247

GOLOWERTHAN
 INQUIRE CAPOPTPRED
 command 247

GREATERTHAN
 INQUIRE CAPDATAPRED
 command 243
 INQUIRE CAPOPTPRED
 command 247
 INQUIRE CAPTURESPEC
 command 250, 251, 252, 253

HTTP
 INQUIRE EPADAPTER
 command 318, 319

ISNOTGREATER
 INQUIRE CAPDATAPRED
 command 243
 INQUIRE CAPOPTPRED
 command 248
 INQUIRE CAPTURESPEC
 command 250, 251, 252, 253

ISNOTLESS
 INQUIRE CAPDATAPRED
 command 243
 INQUIRE CAPOPTPRED
 command 248
 INQUIRE CAPTURESPEC
 command 250, 251, 252, 253

LESSTHAN
 INQUIRE CAPDATAPRED
 command 243
 INQUIRE CAPOPTPRED
 command 248
 INQUIRE CAPTURESPEC
 command 250, 251, 252, 253

MAP
 INQUIRE CAPTURESPEC
 command 253

NO
 INQUIRE URIMAP
 command 555

NOAUTHENTIC
 INQUIRE URIMAP
 command 555

NONE
 INQUIRE CAPTURESPEC
 command 253
 INQUIRE URIMAP
 command 559
 SET URIMAP command 764

NONTRANS
 INQUIRE EPADAPTER
 command 321

NOTAPPLIC
 INQUIRE ATOMSERVICE
 command 228

CVDA values (continued)

PERM
 INQUIRE URIMAP
 command 559

PERMANENT
 SET URIMAP command 764

PGMINIT
 INQUIRE CAPTURESPEC
 command 250

PIPELINE
 INQUIRE URIMAP
 command 560

POSTCMD
 INQUIRE CAPTURESPEC
 command 250

PRECMD
 INQUIRE CAPTURESPEC
 command 250

PROGRAM
 INQUIRE ATOMSERVICE
 command 228
 INQUIRE CAPTURESPEC
 command 254

REGION
 INQUIRE EPADAPTER
 command 319

SERVER
 INQUIRE URIMAP
 command 560

SERVICE
 INQUIRE ATOMSERVICE
 command 226
 INQUIRE CAPTURESPEC
 command 254

STARTSWITH
 INQUIRE CAPDATAPRED
 command 243
 INQUIRE CAPOPTPRED
 command 248
 INQUIRE CAPTURESPEC
 command 250, 251, 252, 253

SYNCHRONOUS
 INQUIRE EPADAPTER
 command 320

SYSTEM
 INQUIRE CAPTURESPEC
 command 250

TDQUEUE
 INQUIRE CAPTURESPEC
 command 254

TEMP
 INQUIRE URIMAP
 command 559

TEMPORARY
 SET URIMAP command 764

TRANS
 INQUIRE EPADAPTER
 command 321

TRANSACTION
 INQUIRE CAPTURESPEC
 command 254

TRANSTART
 INQUIRE EPADAPTER
 command 318

TSQ
 INQUIRE EPADAPTER
 command 318

CVDA values (continued)

TSQUEUE
 INQUIRE ATOMSERVICE
 command 228
 INQUIRE CAPTURESPEC
 command 254

UNKNOWN
 INQUIRE ATOMSERVICE
 command 226

USER
 INQUIRE EPADAPTER
 command 320

USERID
 INQUIRE EPADAPTER
 command 319

WBE
 INQUIRE EPADAPTER
 command 320

WMQ
 INQUIRE EPADAPTER
 command 318

YES
 INQUIRE URIMAP
 command 555

D

DAEOPTION option
 INQUIRE SYSDUMPCODE
 command 438, 713

data table options
 MAXNUMRECS option on SET FILE
 command 662
 TABLE option on SET FILE
 command 662

data types 10

data-area argument
 CICS command format 6

data-areas 6

data-value argument
 CICS command format 6

data-values 6

DATABUFFERS option
 INQUIRE TDQUEUE command 479

DATAFORMAT option
 INQUIRE EPADAPTER
 command 320

DATALOCATION option
 INQUIRE PROGRAM command 410

DATASTREAM option
 INQUIRE TERMINAL command 485

DB2 option
 PERFORM STATISTICS
 command 600

DB2CONN command
 conditions 51

DB2CONN option
 COLLECT STATISTICS command 33
 DISCARD DB2CONN command 175
 INQUIRE SYSTEM command 444

DB2CONN, CREATE command 51

DB2CONN, DISCARD command 175

DB2CONN, INQUIRE command 276

DB2CONN, SET command 624

DB2ENTRY option
 COLLECT STATISTICS command 33
 DISCARD DB2ENTRY command 176

DB2ENTRY, CREATE command 54
 DB2ENTRY, DISCARD command 176
 DB2ENTRY, INQUIRE command 285
 DB2ENTRY, SET command 634
 DB2PLAN option
 INQUIRE TASK command 455
 DB2TRAN option
 DISCARD DB2TRAN command 177
 DB2TRAN, CREATE command 56
 DB2TRAN, DISCARD command 176
 DB2TRAN, INQUIRE command 291
 DB2TRAN, SET command 640
 DDNAME option
 INQUIRE DOCTEMPLATE
 command 303
 INQUIRE TDQUEUE command 479
 DEBUGTOOL option
 INQUIRE SYSTEM command 444
 SET SYSTEM command 717
 DEFINE command
 CSD 133
 defining exits 28
 DEFPAGEHT option
 INQUIRE TERMINAL command 485
 DEFPAGEWD option
 INQUIRE TERMINAL command 485
 DEFSCRNHT option
 INQUIRE TERMINAL command 485
 DEFSCRNWD option
 INQUIRE TERMINAL command 485
 DELETE command
 CSD 137
 DELETE option
 INQUIRE FILE command 337
 SET FILE command 662
 DELETSHIPED, INQUIRE
 command 294
 DELETSHIPED, PERFORM
 command 585
 DELETSHIPED, SET command 641
 DEREGERROR value 471
 DEREGISTERED option
 SET VTAM command 765
 DEREGISTERED value 471
 DEVICE option
 INQUIRE TERMINAL command 485
 DFLTUSER option
 INQUIRE SYSTEM command 445
 DISABLE PROGRAM command 167
 conditions 169
 examples for global user exits 170
 DISABLED CVDA value
 INQUIRE AUTOINSTALL
 command 231
 DISCARD ATOMSERVICE
 command 170
 conditions 170
 DISCARD AUTINSTMODEL
 command 171
 conditions 171
 DISCARD BUNDLE command 172
 conditions 172
 DISCARD commands
 BUNDLE 172
 CONNECTION 172
 CORBASERVER 174
 DJAR 177
 DISCARD commands (*continued*)
 JVMSERVER 183
 PIPELINE 187
 TERMINAL 193
 WEBSERVICE 198
 DISCARD CONNECTION
 command 172
 conditions 173
 DISCARD CORBASERVER
 command 174
 conditions 174
 DISCARD DB2CONN command 175
 conditions 175
 DISCARD DB2ENTRY command 176
 conditions 176
 DISCARD DB2TRAN command 176
 conditions 177
 DISCARD DJAR command 177
 conditions 178
 DISCARD DOCTEMPLATE
 command 178
 conditions 178
 DISCARD ENQMODEL command 179
 conditions 179
 DISCARD FILE command 180
 conditions 180
 DISCARD IPCONN command 181
 conditions 181
 DISCARD JOURNALMODEL
 command 181
 conditions 182
 DISCARD JOURNALNAME
 command 182
 conditions 183
 DISCARD JVMSERVER command 183
 conditions 184
 DISCARD LIBRARY command 184
 conditions 184
 DISCARD MQCONN command 185
 conditions 186
 DISCARD option
 CREATE CONNECTION
 command 48
 CREATE TERMINAL command 108
 DISCARD PARTNER command 186
 conditions 186
 DISCARD PIPELINE command 187
 conditions 187
 DISCARD PROCESSTYPE
 command 188
 conditions 188
 DISCARD PROFILE command 188
 conditions 189
 DISCARD PROGRAM command 189
 conditions 190
 DISCARD REQUESTMODEL
 command 190
 conditions 191
 DISCARD TCPIPSERVICE
 command 191
 conditions 191
 DISCARD TDQUEUE command 192
 conditions 192
 DISCARD TERMINAL command 193
 conditions 194
 DISCARD TRANCLASS command 194
 conditions 195
 DISCARD TRANSACTION
 command 195
 conditions 196
 DISCARD TSMODEL command 196
 conditions 197
 DISCARD URIMAP command 197
 conditions 197
 DISCARD WEBSERVICE command 198
 conditions 198
 discarding resources
 resource definitions 27, 171
 DISCONNECT command
 CSD 139
 DISCREQST option
 INQUIRE TERMINAL 736
 INQUIRE TERMINAL command 485
 DISPATCHABLE option
 INQUIRE TASK LIST command 463
 DISPATCHER option
 COLLECT STATISTICS command 33
 PERFORM STATISTICS
 command 601
 DISPATCHER, INQUIRE command 295
 DISPATCHER, SET command 643
 DISPOSITION option
 INQUIRE FILE command 337
 INQUIRE TDQUEUE command 479
 SET FILE command 662
 DJAR option
 CREATE DJAR command 59
 DISCARD DJAR command 178
 INQUIRE BEAN command 232
 INQUIRE DJAR command 300
 DJAR, CREATE command 58
 DJAR, DISCARD command 177
 DJAR, INQUIRE command 298
 DJAR, PERFORM 586
 DJARDIR option
 INQUIRE CORBASERVER
 command 273
 DNAME option
 INQUIRE ASSOCIATION
 command 212
 INQUIRE ASSOCIATION LIST
 command 222
 DNAMLEN option
 INQUIRE ASSOCIATION LIST
 command 222
 DNSGROUP option
 INQUIRE TCPIPSERVICE
 command 470
 DNSSTATUS option
 INQUIRE TCPIPSERVICE
 command 470
 SET TCPIPSERVICE command 727
 DOCTEMPLATE CVDA value
 EXTRACT STATISTICS
 command 209
 DOCTEMPLATE option
 INQUIRE DOCTEMPLATE
 command 304
 PERFORM STATISTICS
 command 601
 SET DOCTEMPLATE command 646
 DOCTEMPLATE, DISCARD
 command 178

DOCTEMPLATE, INQUIRE
command 301

DOCTEMPLATE, SET command 646

document templates
caching 646

DPLLIMIT option
INQUIRE MONITOR command 382
SET MONITOR command 695

DSALIMIT option
INQUIRE SYSTEM command 445
SET SYSTEM command 717

DSANAME option
INQUIRE STORAGE command 433
INQUIRE SUBPOOL command 437

DSNAME option
INQUIRE DOCTEMPLATE
command 304
INQUIRE DSNAME command 308
INQUIRE FILE command 337
INQUIRE LIBRARY command 376
INQUIRE TDQUEUE command 480
SET DSNAME command 651
SET FILE command 662

DSNAME, INQUIRE command 306

DSNAME, SET command 647

DSNAMELIST option
INQUIRE LIBRARY command 376

DSRTPROGRAM option
INQUIRE SYSTEM command 445
SET SYSTEM command 717

DTIMEOUT option
INQUIRE TASK command 455
INQUIRE TRANSACTION
command 520

DTRPROGRAM option
INQUIRE SYSTEM command 445
SET SYSTEM command 717

DUALCASEST option
INQUIRE TERMINAL command 493

dump data sets 312

DUMP option
PERFORM SHUTDOWN
command 597

DUMP, PERFORM command 587

DUMPCODE option
PERFORM DUMP command 587

DUMPDS, INQUIRE command 312

DUMPDS, SET command 656

DUMPID option
PERFORM DUMP command 587

DUMPING option
INQUIRE SYSTEM command 445
INQUIRE TASK command 455
INQUIRE TRANSACTION
command 520
SET SYSTEM command 717
SET TRANSACTION command 757

DUMPSCOPE option
INQUIRE SYSDUMPCODE
command 439, 713
INQUIRE TRANDUMPCODE
command 515
SET TRANDUMPCODE
command 754

DURATION option
INQUIRE UOWENQ command 547

DYNAMSTATUS option
INQUIRE PROGRAM command 410

E

ECDSASIZE option
INQUIRE SYSTEM command 445

ECI value
INQUIRE TCPIPSERVICE
command 474

EDSALIMIT option
INQUIRE SYSTEM command 445
SET SYSTEM command 717

EDSASIZE option
INQUIRE SYSTEM command 445

ELEMENT option
INQUIRE STORAGE command 433

ELEMENTLIST option
INQUIRE MVSTCB command 391
INQUIRE STORAGE command 433

EMITMODE option
INQUIRE EPADAPTER
command 320

EMPTY option
SET FILE command 662

EMPTYSTATUS option
INQUIRE FILE command 338
INQUIRE TDQUEUE command 480
SET FILE command 662

ENABLE PROGRAM command 199
conditions 203
examples for global user exits 204
examples for task-related user
exits 205

ENABLED CVDA value
INQUIRE AUTOINSTALL
command 231

ENABLEDCOUNT option
INQUIRE BUNDLE command 237

ENABLESTATUS option
INQUIRE ATOMSERVICE
command 227
INQUIRE AUTOINSTALL
command 231
INQUIRE BUNDLE command 237
INQUIRE BUNDLEPART
command 239
INQUIRE CORBASERVER
command 273
INQUIRE EPADAPTER
command 320
INQUIRE EVENTBINDING
command 323
INQUIRE FILE command 338
INQUIRE HOST command 347
INQUIRE LIBRARY command 376
INQUIRE TDQUEUE command 480
INQUIRE URIMAP command 556
SET ATOMSERVICE command 608
SET BUNDLE command 612
SET CORBASERVER command 623
SET EPADAPTER command 660
SET EVENTBINDING command 661
SET FILE command 662
SET HOST command 676
SET LIBRARY command 690
SET TDQUEUE command 730

ENABLESTATUS option (*continued*)
SET URIMAP command 763

END condition
INQUIRE ATOMSERVICE
command 228
INQUIRE CAPDATAPRED
command 241
INQUIRE CAPINFOSRCE
command 246
INQUIRE CAPOPTPRED
command 248
INQUIRE CAPTURESPEC
command 254
INQUIRE EPADAPTER
command 321
INQUIRE EVENTBINDING
command 324
INQUIRE HOST command 348
INQUIRE JVMSERVER
command 373
INQUIRE URIMAP command 561

ENDAFFINITY, PERFORM
command 590

ENDBRGROUP command
CSD 139

ENDBRLIST command
CSD 140

ENDBRRSRCE command
CSD 141

ENDOFDAY option
INQUIRE STATISTICS command 431
SET STATISTICS command 710

ENDOFDAYHRS option
INQUIRE STATISTICS command 431
SET STATISTICS command 710

ENDOFDAYMINS option
INQUIRE STATISTICS command 431
SET STATISTICS command 710

ENDOFDAYSECS option
INQUIRE STATISTICS command 431
SET STATISTICS command 711

ENDPOINT option
INQUIRE WEBSERVICE
command 568

ENQ, INQUIRE command 313

ENQFAILS option
INQUIRE UOWENQ command 547

ENQMODEL option
CREATE ENQMODEL command 63
DISCARD ENQMODEL
command 179
INQUIRE ENQMODEL
command 316

ENQMODEL, DISCARD command 179

ENQMODEL, INQUIRE command 314

ENQNAME option
INQUIRE ENQMODEL
command 316

ENQSCOPE option
INQUIRE ENQMODEL
command 316
INQUIRE UOWENQ command 547

ENQUEUE option
COLLECT STATISTICS command 33
PERFORM STATISTICS
command 601

ENTRY option
 ENABLE PROGRAM command 200
 INQUIRE EXITPROGRAM
 command 329
 ENTRYNAME option
 DISABLE PROGRAM command 167
 ENABLE PROGRAM command 200
 EXTRACT EXIT command 205
 INQUIRE EXITPROGRAM
 command 329
 RESYNC command 607
 ENTRYNAME, RESYNC command 605
 ENTRYPOINT option
 INQUIRE PROGRAM command 410
 EPADAPTER CVDA value
 EXTRACT STATISTICS
 command 209
 EPADAPTER option
 INQUIRE EPADAPTER
 command 323
 PERFORM STATISTICS
 command 601
 SET EPADAPTER command 660
 EPADAPTER, INQUIRE command 317
 EPADAPTER, SET command 659
 EPSTATUS option
 INQUIRE EVENTPROCESS 325
 SET EVENTPROCESS command 662
 ERDSASIZE option
 INQUIRE SYSTEM command 445
 ERROROPTION option
 INQUIRE TDQUEUE command 480
 ESM (external security manager) 2
 ESMRESP option
 PERFORM SECURITY REBUILD
 command 595
 EUDSASIZE option
 INQUIRE SYSTEM command 445
 EVENTBINDING CVDA value
 EXTRACT STATISTICS
 command 209
 EVENTBINDING option
 INQUIRE CAPTURESPEC
 command 252
 INQUIRE EVENTBINDING
 command 324
 PERFORM STATISTICS
 command 601
 SET EVENTBINDING command 661
 EVENTBINDING, INQUIRE
 command 322
 EVENTBINDING, SET command 660
 EVENTNAME option
 INQUIRE CAPTURESPEC
 command 252
 EVENTPROCESS CVDA value
 EXTRACT STATISTICS
 command 210
 EVENTPROCESS option
 PERFORM STATISTICS
 command 601
 EVENTPROCESS, INQUIRE
 command 324
 EVENTPROCESS, SET command 661
 EXCEPTCLASS option
 INQUIRE MONITOR command 382
 SET MONITOR command 695

EXCI option
 INQUIRE EXCI command 325
 EXCI, INQUIRE command 325
 EXCLUSIVE option
 INQUIRE FILE command 338
 SET FILE command 662
 EXEC CICS commands
 format 2
 Function codes 825
 Response Codes 825
 EXEC CICS CREATE
 RESP2 values 837
 EXEC CICS SPI commands
 Threadsafe 863
 EXECKEY option
 INQUIRE JVM command 366
 INQUIRE PROGRAM command 410
 PERFORM JVMPOOL command 591
 execution diagnostic facility transaction
 (CEDF) 1
 EXECUTIONSET option
 INQUIRE PROGRAM command 410
 SET PROGRAM command 706
 exit names 29
 EXIT option
 DISABLE PROGRAM command 168
 ENABLE PROGRAM command 200
 INQUIRE DOCTEMPLATE
 command 305
 INQUIRE EXITPROGRAM
 command 329
 exit-related commands 28
 EXIT, EXTRACT command 205
 EXITALL option
 DISABLE PROGRAM command 168
 EXITPGM option
 INQUIRE DOCTEMPLATE
 command 304
 EXITPROGRAM option
 INQUIRE EXITPROGRAM
 command 329
 EXITPROGRAM, INQUIRE
 command 326
 exits
 defining 28
 EXITTRACING option
 INQUIRE CONNECTION
 command 263
 INQUIRE TERMINAL command 493
 SET CONNECTION command 616
 SET NETNAME command 701
 SET TERMINAL command 736
 EXPIRYINT option
 INQUIRE TSQNAME command 532
 INQUIRE TSQUEUE command 532
 EXTENDEDSSST option
 INQUIRE TERMINAL command 493
 external security manager (ESM) 2
 EXTRACT EXIT command 205
 conditions 206
 EXTRACT STATISTICS
 conditions 211
 EXTRACT STATISTICS command 207

F

FACILITY option
 INQUIRE TASK command 455
 FACILITYLIKE option
 INQUIRE TRANSACTION
 command 521
 FACILITYTYPE option
 INQUIRE TASK command 455
 FACILNAME option
 INQUIRE ASSOCIATION
 command 212
 FACILTYE option
 INQUIRE ASSOCIATION
 command 212
 FEPI option
 PERFORM STATISTICS
 command 601
 FIELDLENGTH option
 INQUIRE CAPDATAPRED
 command 242
 INQUIRE CAPINFOSRCE
 command 245
 FIELDOFFSET option
 INQUIRE CAPDATAPRED
 command 242
 INQUIRE CAPINFOSRCE
 command 245
 FILE option
 COLLECT STATISTICS command 33
 CREATE FILE command 63
 DISCARD FILE command 180
 INQUIRE DOCTEMPLATE
 command 304, 305
 INQUIRE FILE command 339
 INQUIRE PROCESSTYPE
 command 406
 PERFORM STATISTICS
 command 601
 SET FILE command 662
 FILE, DISCARD command 180
 FILE, INQUIRE command 332
 FILE, SET command 662
 FILECOUNT option
 INQUIRE DSNAME command 308
 FILELIMIT option
 INQUIRE MONITOR command 382
 SET MONITOR command 695
 filename argument, CICS command
 format 6
 FILENAME option
 INQUIRE CAPDATAPRED
 command 242
 INQUIRE CAPINFOSRCE
 command 245
 FILTERVALUE option
 INQUIRE CAPDATAPRED
 command 242
 INQUIRE CAPOPTPRED
 command 247
 FLAGSET option
 INQUIRE TRACETYPE
 command 511
 SET TRACETYPE command 750
 FLENGTH option
 INQUIRE STORAGE command 434
 INQUIRE TSQNAME command 532
 INQUIRE TSQUEUE command 532

FMHPARMST option
 INQUIRE TERMINAL command 493

FMHSTATUS option
 INQUIRE REQID command 423

FORCECANCEL option
 SET CONNECTION command 617
 SET IPCONN command 678

FORCEQR option
 INQUIRE SYSTEM command 446
 SET SYSTEM command 718

format rules 5

FORMATEDF option
 DISABLE PROGRAM command 168
 ENABLE PROGRAM command 200

FORMATEDFST option
 INQUIRE EXITPROGRAM
 command 329

FORMFEEDST option
 INQUIRE TERMINAL command 494

FREQUENCY option
 INQUIRE MONITOR command 382
 SET MONITOR command 695

FREQUENCYHRS option
 INQUIRE MONITOR command 383
 SET MONITOR command 696

FREQUENCYMIN option
 INQUIRE MONITOR command 383
 SET MONITOR command 696

FREQUENCYSEC option
 INQUIRE MONITOR command 383
 SET MONITOR command 696

Function codes
 of EXEC CICS commands 825

function shipping, not available for SP
 commands 1

FWDRECOVLOG option
 INQUIRE DSNAME 308

FWDRECOVLSN option
 INQUIRE DSNAME 308

FWDRECSTATUS option
 INQUIRE FILE command 339

G

GAENTRYNAME option
 ENABLE PROGRAM command 200
 INQUIRE EXITPROGRAM
 command 330

GALENGTH option
 ENABLE PROGRAM command 201
 EXTRACT EXIT command 206
 INQUIRE EXITPROGRAM
 command 330

GARBAGEINT
 CEMT INQUIRE WEB 565
 SET WEB command 769

GASET option
 EXTRACT EXIT command 206

GAUSECOUNT option
 INQUIRE EXITPROGRAM
 command 330

GCHARS option
 INQUIRE TERMINAL command 494

GCODES option
 INQUIRE TERMINAL command 494

GETNEXTGROUP command
 CSD 142

GETNEXTLIST command
 CSD 143

GETNEXTRSRCE command
 CSD 143

GMMLENGTH option
 INQUIRE SYSTEM command 446
 SET SYSTEM command 718

GMMTEXT option
 INQUIRE SYSTEM command 446
 SET SYSTEM command 718

GMMTRANID option
 INQUIRE SYSTEM command 446

GRNAME option
 INQUIRE CONNECTION
 command 263
 INQUIRE VTAM command 562

GROUP option
 DSNCRCT macro 636

GRPCRITICAL option
 INQUIRE TCPIP SERVICE
 command 471

GRSTATUS option
 INQUIRE VTAM command 562

GTFSTATUS option
 INQUIRE TRACEDEST
 command 506
 SET TRACEDEST command 744

H

HFORMST option
 INQUIRE TERMINAL command 494

HFSFILE option
 INQUIRE DOCTEMPLATE
 command 304, 305
 INQUIRE URIMAP command 557

HFSFILR option
 INQUIRE DJAR command 300

HFSNAME option
 INQUIRE JVMPROFILE
 command 369

HIGHLIGHTST option
 INQUIRE TERMINAL command 494

HOLDSTATUS option
 INQUIRE PROGRAM command 410

HOST option
 INQUIRE CORBASERVER
 command 273
 INQUIRE HOST command 347, 675
 INQUIRE IPCONN command 348
 INQUIRE TCPIP SERVICE
 command 471
 INQUIRE UOWLINK command 551
 INQUIRE URIMAP command 557

HOSTCODEPAGE option
 INQUIRE URIMAP command 557

HOSTTYPE option
 INQUIRE CORBASERVER
 command 274
 INQUIRE IPCONN command 348
 INQUIRE TCPIP SERVICE
 command 471
 INQUIRE URIMAP command 557

HOURS option
 INQUIRE REQID command 423

HTTP value
 INQUIRE TCPIP SERVICE
 command 474

I

IDENTIFIER option
 INQUIRE TASK command 456

IDLIST option
 RESYNC command 607

IDLISTLENGTH option
 RESYNC command 607

IDNTYCLASS option
 INQUIRE MONITOR command 383
 SET MONITOR command 696

IDPROP option
 INQUIRE IPCONN command 348

IGNORE (null values) 14

IIOF value
 INQUIRE TCPIP SERVICE
 command 474

ILLOGIC condition
 INQUIRE ATOMSERVICE
 command 228
 INQUIRE CAPDATAPRED
 command 241
 INQUIRE CAPINFOSRCE
 command 246
 INQUIRE CAPOPTPRED
 command 248
 INQUIRE CAPTURESPEC
 command 254
 INQUIRE EPADAPTER
 command 321
 INQUIRE EVENTBINDING
 command 324
 INQUIRE HOST command 348
 INQUIRE JVMSERVER
 command 373
 INQUIRE URIMAP command 561

IMMCLOSE value
 INQUIRE TCPIP SERVICE
 command 473
 SET TCPIP command 725
 SET TCPIP SERVICE command 727

IMMCLOSING value
 INQUIRE TCPIP command 466
 INQUIRE TCPIP SERVICE
 command 473

IMMEDIATE option
 PERFORM SHUTDOWN
 command 597

INDIRECTNAME option
 INQUIRE TDQUEUE command 481

INDOUBT option
 INQUIRE TASK command 456
 INQUIRE TDQUEUE command 481
 INQUIRE TRANSACTION
 command 521

INDOUBTMINS option
 INQUIRE TASK command 456
 INQUIRE TRANSACTION
 command 521

INDOUBTST option
 INQUIRE EXITPROGRAM
 command 330

INDOUBTWAIT option
 ENABLE PROGRAM command 201
 INQUIRE TDQUEUE command 481
 INQUIRE TRANSACTION
 command 456, 521
 INITIALDDS option
 INQUIRE DUMPDS command 313
 SET DUMPDS command 657
 INITIALIZE option
 PERFORM CLASSCACHE
 command 581
 PERFORM JVMPOOL command 591
 INITSTATUS option
 INQUIRE SYSTEM command 446
 INITUSERID option
 INQUIRE ASSOCIATION
 command 212
 INQUIRE and SET commands
 examples
 Assembler 10
 C 9
 COBOL 9
 PL/I 10
 null values 14
 INQUIRE ASSOCIATION command 212
 conditions 212
 INQUIRE ASSOCIATION LIST
 command 222
 conditions 222
 INQUIRE ATOMSERVICE
 command 225
 conditions 228
 INQUIRE AUTINSTMODEL
 command 229
 conditions 229
 INQUIRE AUTOINSTALL
 command 230
 conditions 231
 INQUIRE BEAN command 232
 conditions 232
 INQUIRE BRFACILITY command 233
 conditions 235
 INQUIRE BUNDLE command 235
 conditions 238
 INQUIRE BUNDLEPART command 239
 conditions 239
 INQUIRE CAPDATAPRED
 command 241
 conditions 241
 INQUIRE CAPINFOSRCE
 command 244
 conditions 246
 INQUIRE CAPOPTPRED command 246
 conditions 248
 INQUIRE CAPTURESPEC
 command 249
 conditions 254
 INQUIRE CFDTPPOOL command 254
 INQUIRE CLASSCACHE command 256
 conditions 258
 INQUIRE command, browse
 conditions 23
 INQUIRE commands
 ASSOCIATION 212
 ASSOCIATION LIST 222
 AUTINSTMODEL 229
 AUTOINSTALL 230
 BEAN 232
 BRFACILITY 233
 BUNDLE 235
 BUNDLEPART 239
 CAPDATAPRED 241
 CAPINFOSRCE 244
 CAPOPTPRED 246
 CAPTURESPEC 249
 CFDTPPOOL 254
 CLASSCACHE 256
 CONNECTION 258
 CORBASERVER 271
 DB2CONN 276
 DB2ENTRY 285
 DB2TRAN 291
 DELETSHIPPED 294
 DISPATCHER 295
 DJAR 298
 DOCTEMPLATE 301
 DSNAME 306
 DUMPDS 312
 ENQ 313
 ENQMODEL 314
 EPADAPTER 317
 EVENTBINDING 322
 EVENTPROCESS 324
 EXCI 325
 EXITPROGRAM 326
 FILE 332
 IPCONN 348
 IPFACILITY 357
 IRC 358
 JOURNALNUM 364
 JVM 365
 JVMPOOL 367
 JVMPROFILE 368
 JVMSERVER 370
 LIBRARY 374
 MODENAME 378
 MONITOR 380
 MQCONN 384
 MQINI 388
 MVSTCB 390
 NETNAME 392
 OSGIBUNDLE 393
 OSGISERVICE 397
 PARTNER 398
 PIPELINE 400
 PROCESSTYPE 405, 703
 PROFILE 408
 PROGRAM 410
 REQID 421
 STATISTICS 430
 STORAGE 432
 SUBPOOL 436
 SYSDUMPCODE 437
 SYSTEM 440
 TASK 452
 TASK LIST 463
 TCLASS 464
 TCPIP 466
 TCPIPSERVICE 467
 TDQUEUE 475
 TEMPSTORAGE 485
 TERMINAL 485
 TRACEDEST 505
 TRACEFLAG 507
 TRACETYPE 509
 TRANCLASS 512
 TRANDUMPCODE 514
 TRANSACTION 517
 TSMODEL 527
 TSPOOL 530
 TSQUEUE 531
 UOW 535
 UOWDSNFAIL 539
 UOWENQ 544
 UOWLINK 550
 VOLUME 561
 WEB 564
 WORKREQUEST 571
 z/OS Communications Server 561
 INQUIRE CONNECTION
 command 258
 conditions 267
 INQUIRE CORBASERVER
 command 271
 conditions 275, 301
 INQUIRE DB2CONN command 276
 conditions 276
 INQUIRE DB2ENTRY command 285
 conditions 291
 INQUIRE DB2TRAN command 291
 conditions 293
 INQUIRE DELETSHIPPED
 command 294
 conditions 295
 INQUIRE DISPATCHER command 295
 conditions 298
 INQUIRE DJAR command 298
 INQUIRE DOCTEMPLATE
 command 301
 conditions 305
 INQUIRE DSNAME command 306
 conditions 311, 544
 INQUIRE DUMPDS command 312
 INQUIRE ENQ command 313
 INQUIRE ENQMODEL command 314
 conditions 317
 INQUIRE EPADAPTER command 317
 conditions 321
 INQUIRE EVENTBINDING
 command 322
 conditions 324
 INQUIRE EVENTPROCESS
 command 324
 conditions 325
 INQUIRE EXCI command 325
 conditions 326
 INQUIRE EXITPROGRAM
 command 326
 conditions 332
 INQUIRE FILE command 332
 INQUIRE HOST command 347
 conditions 348
 INQUIRE IPCONN command 348
 conditions 348
 INQUIRE IPFACILITY command 357
 INQUIRE IRC command 358
 conditions 359
 INQUIRE JOURNALMODEL
 command 359

INQUIRE JOURNALMODEL command
(continued)
conditions 362

INQUIRE JOURNALNAME
command 363
conditions 364

INQUIRE JOURNALNUM
command 364

INQUIRE JVM command 365
conditions 366, 369

INQUIRE JVMPOOL command 367
conditions 368

INQUIRE JVMPROFILE command 368

INQUIRE JMVSERVER command 370
conditions 373

INQUIRE LIBRARY command 374

INQUIRE MODENAME command 378
conditions 378, 436

INQUIRE MONITOR command 380
conditions 384

INQUIRE MQCONN command 384
conditions 384

INQUIRE MQINI command 388
conditions 390

INQUIRE MVSTCB command 390
conditions 392

INQUIRE NETNAME command 392

INQUIRE OSGIBUNDLE command 393
conditions 396

INQUIRE OSGISERVICE command 397
conditions 398

INQUIRE PARTNER command 398
conditions 399

INQUIRE PIPELINE command 400
conditions 404

INQUIRE PROCESSTYPE command 405
conditions 407, 704

INQUIRE PROFILE command 408
conditions 409
PROFILE 409

INQUIRE PROGRAM command 410
conditions 410

INQUIRE REQID command 421
conditions 425

INQUIRE REQUESTMODEL
command 426
conditions 428

INQUIRE rrms command
conditions 429

INQUIRE RRMS command 429

INQUIRE STATISTICS command 430
conditions 432

INQUIRE STORAGE command 432
conditions 434

INQUIRE STREAMNAME
command 435

INQUIRE SUBPOOL command
conditions 437
SUBPOOL 436

INQUIRE SYSDUMPCODE
command 437
conditions 440

INQUIRE SYSTEM command 440
conditions 452

INQUIRE TASK command 452, 455
conditions 463

INQUIRE TASK LIST command 463

INQUIRE TASK LIST command
(continued)
conditions 463

INQUIRE TCLASS command 464
conditions 465

INQUIRE TCPIP command 466
conditions 467

INQUIRE TCPIPSERVICE 470, 471

INQUIRE TCPIPSERVICE command 467
conditions 474

INQUIRE TDQUEUE command 475
conditions 484

INQUIRE TEMPSTORAGE
command 485
conditions 485

INQUIRE TERMINAL command 485
conditions 485

INQUIRE TRACEDEST command 505
conditions 507

INQUIRE TRACEFLAG command 507
conditions 509

INQUIRE TRACETYPE command 509
conditions 511

INQUIRE TRANCLASS command 512
conditions 514

INQUIRE TRANDUMPCODE
command 514
conditions 517

INQUIRE TRANSACTION
command 517
conditions 526

INQUIRE TSMODEL command 527
conditions 529

INQUIRE TSPOOL command 530
conditions 531

INQUIRE TSQNAME command 531
conditions 534

INQUIRE TSQUEUE command
conditions 534
TSQNAME 531

INQUIRE UOW command 535
conditions 539

INQUIRE UOWDSNFAIL command 539

INQUIRE UOWENQ command 544
conditions 549

INQUIRE UOWLINK command 550
conditions 553

INQUIRE URIMAP command 554
conditions 561

INQUIRE VOLUME command 561
conditions 561

INQUIRE VTAM command 561
conditions 564

INQUIRE WEB command
conditions 565

INQUIRE WEBSERVICE command 565
conditions 571

INQUIRE WORKREQUEST command
conditions 575
WORKREQUEST 571

INQUIRE XMLTRANSFORM
command 575

INQUIREGROUP command
CSD 146

INQUIRELISTcommand
CSD 147

INQUIRERSRCE command
CSD 147

inquiry commands 19

INSTALL command
CSD 150

integer-expr argument, CICS command
format 6

INTERFACE option
INQUIRE REQUESTMODEL
command 428

INTERVAL option
INQUIRE REQID command 423
INQUIRE STATISTICS command 431
SET STATISTICS command 711

INTERVALHRS option
INQUIRE STATISTICS command 431
SET STATISTICS command 711

INTERVALMINS option
INQUIRE STATISTICS command 431
SET STATISTICS command 711

INTERVALSECS option
INQUIRE STATISTICS command 431
SET STATISTICS command 711

INTFACETYPE option
INQUIRE REQUESTMODEL
command 428

INTSTATUS option
INQUIRE TRACEDEST
command 506
SET TRACEDEST command 744

INVREQ condition
DISCARD ATOMSERVICE
command 170
DISCARD URIMAP command 197
INQUIRE CAPDATAPRED
command 241
INQUIRE CAPINFOSRCE
command 246
INQUIRE CAPOPTPRED
command 248
INQUIRE CAPTURESPEC
command 254
INQUIRE HOST command 348
SET ATOMSERVICE command 608
SET EPADAPTER command 660
SET EVENTBINDING command 661
SET EVENTPROCESS command 662
SET URIMAP command 764
SET WEBSERVICE command 770

IOTYPE option
INQUIRE TDQUEUE command 482

IPADDRESS option
INQUIRE TCPIPSERVICE
command 472

IPCONN
SET IPCONN command 677

IPCONN CVDA value
EXTRACT STATISTICS
command 210

IPCONN option
CREATE IPCONN command 69
DISCARD IPCONN command 181
INQUIRE ASSOCIATION
command 212
INQUIRE IPCONN command 348
PERFORM STATISTICS
command 601

IPCONN, INQUIRE command 348
 IPCONN, SET command 676
 IPCONNs 348
 IPFACILITIES option
 INQUIRE TASK command 457
 IPFACILITY, INQUIRE command 357
 IPFAMILY option
 INQUIRE ASSOCIATION
 command 212
 INQUIRE CORBASERVER
 command 274
 INQUIRE IPCONN command 348
 INQUIRE TCIPSERVICE
 command 472
 INQUIRE URIMAP command 558
 IPFLISTSIZ option
 INQUIRE TASK command 457
 IPIC value
 INQUIRE TCIPSERVICE
 command 474
 IPRESOLVED option
 INQUIRE CORBASERVER
 command 274
 INQUIRE IPCONN command 348
 INQUIRE TCIPSERVICE
 command 472
 INQUIRE URIMAP command 558
 IRC, INQUIRE command 358
 IRC, SET command 681
 ISOLATEST option
 INQUIRE TASK command 457
 INQUIRE TRANSACTION
 command 522
 ITEMNAME option
 INQUIRE CAPINFOSRCE
 command 245

J

JNDIPREFIX option
 INQUIRE CORBASERVER
 command 275
 JOBNAME option
 INQUIRE SYSTEM command 446
 JOURNALMODEL option
 CREATE JOURNALMODEL
 command 71
 JOURNALMODEL, CREATE
 command 70
 JOURNALMODEL, DISCARD
 command 181
 JOURNALMODEL, INQUIRE
 command 359
 JOURNALNAME option
 COLLECT STATISTICS command 33
 PERFORM STATISTICS
 command 601
 JOURNALNAME, DISCARD
 command 182
 JOURNALNAME, INQUIRE
 command 363
 JOURNALNAME, SET command 683
 JOURNALNUM option
 COLLECT STATISTICS command 33
 INQUIRE FILE command 340
 PERFORM STATISTICS
 command 601

JOURNALNUM, INQUIRE
 command 364
 JOURNALNUM, SET command 685
 JVM, INQUIRE command 365
 JVMCLASS option
 INQUIRE PROGRAM command 410
 SET PROGRAM command 706
 JVMCOUNT option
 PERFORM JVMPOOL command 591
 JVMLVnTRACE option
 INQUIRE JVMPOOL command 367
 SET JVMPOOL command 686
 JVMPOOL option
 COLLECT STATISTICS command 33
 PERFORM STATISTICS
 command 601
 JVMPOOL, INQUIRE command 367
 JVMPOOL, PERFORM command 591
 JVMPOOL, SET command 685
 JVMPROFILE option
 COLLECT STATISTICS command 33
 INQUIRE PROGRAM command 410
 PERFORM JVMPOOL command 591
 PERFORM STATISTICS
 command 601
 SET PROGRAM command 706
 JVMPROFILE, INQUIRE command 368
 JVMPROGRAM option
 COLLECT STATISTICS command 33
 PERFORM STATISTICS
 command 601
 JVMSERVER CVDA value
 EXTRACT STATISTICS
 command 210
 JVMSERVER option
 CREATE JVMSERVER command 73
 INQUIRE PROGRAM command 410
 PERFORM STATISTICS
 command 601
 JVMSERVER, CREATE command 72
 JVMSERVER, DISCARD command 183
 JVMSERVER, INQUIRE command 370
 JVMSERVER, SET command 686
 JVMUSERTRACE option
 INQUIRE JVMPOOL command 367
 SET JVMPOOL command 686

K

KATAKANAST option
 INQUIRE TERMINAL command 494
 KEPTIME option
 INQUIRE BRFACILITY
 command 233
 KEYLENGTH option
 INQUIRE FILE command 340
 SET FILE command 662
 KEYPOSITION option
 INQUIRE FILE command 340

L

label argument, CICS command
 format 6
 LANGDEDUCED option
 INQUIRE PROGRAM command 410

LANGUAGE option
 INQUIRE PROGRAM command 410
 LASTMODTIME option
 INQUIRE DJAR command 300
 INQUIRE WEBSERVICE
 command 568
 LASTRESET option
 COLLECT STATISTICS command 33
 EXTRACT STATISTICS
 command 209
 LASTRESETHRS option
 COLLECT STATISTICS command 33
 EXTRACT STATISTICS
 command 209
 LASTRESETMIN option
 COLLECT STATISTICS command 33
 EXTRACT STATISTICS
 command 209
 LASTRESETSEC option
 COLLECT STATISTICS command 33
 EXTRACT STATISTICS
 command 209
 LASTUSEDINT option
 INQUIRE TSQNAME command 533
 INQUIRE TSQUEUE command 533
 LENGTH option
 default (PL/I) 11
 INQUIRE PROGRAM command 410
 INQUIRE REQID command 424
 LENGTHLIST option
 INQUIRE MVSTCB command 391
 INQUIRE STORAGE command 434
 LIBRARY CVDA value
 EXTRACT STATISTICS
 command 210
 LIBRARY option
 CREATE LIBRARY command 75
 DISCARD LIBRARY command 184
 INQUIRE LIBRARY command 376
 INQUIRE PROGRAM command 410
 PERFORM STATISTICS
 command 601
 SET LIBRARY command 689
 LIBRARY, DISCARD command 184
 LIBRARY, INQUIRE command 374
 LIBRARY, SET command 689
 LIBRARYDSN option
 INQUIRE PROGRAM command 410
 LIGHTPENST option
 INQUIRE TERMINAL command 495
 LINK option
 INQUIRE UOW command 537
 INQUIRE UOWLINK command 551
 LINKAUTH option
 INQUIRE IPCONN command 348
 LINKEDITMODE option
 ENABLE PROGRAM command 201
 LINKSYSNET option
 INQUIRE BRFACILITY
 command 234
 LINKSYSTEM option
 INQUIRE BRFACILITY
 command 234
 INQUIRE CONNECTION
 command 264
 INQUIRE TERMINAL command 495

LISTENERPORT option
 INQUIRE WORKREQUEST
 command 573
 LISTSIZ option
 INQUIRE ASSOCIATION LIST
 command 222
 INQUIRE TASK LIST command 464
 literal constants 13
 LOADPOINT option
 INQUIRE PROGRAM command 410
 LOADTYPE option
 INQUIRE FILE command 340
 SET FILE command 662
 LOCATION option
 INQUIRE CAPDATAPRED
 command 242
 INQUIRE CAPINFOSRCE
 command 245
 INQUIRE TSQNAME command 533
 INQUIRE TSQUEUE command 533
 INQUIRE URIMAP command 558
 SET URIMAP command 764
 LOCK command
 CSD 153
 LOGDEFER option
 INQUIRE SYSTEM command 446
 SET SYSTEM command 718
 LOSTLOCKS option
 INQUIRE DSNAME 309
 LPASTATUS option
 INQUIRE PROGRAM command 410
 LSRPOOL option
 COLLECT STATISTICS command 33
 CREATE LSRPOOL command 79
 PERFORM STATISTICS
 command 601
 LSRPOOL, CREATE command 77
 LSRPOOLID 77
 LSRPOOLID option
 INQUIRE FILE command 341
 SET FILE command 662
 LSRPOOLNUM option
 SET FILE command 662
 LUNAME option
 INQUIRE ASSOCIATION
 command 212

M

MAPNAME option
 INQUIRE TERMINAL command 495
 SET TERMINAL command 736
 MAPPINGLEVEL option
 INQUIRE WEBSERVICE 569
 MAPPINGRNUM option
 INQUIRE WEBSERVICE 569
 MAPPINGVNUM option
 INQUIRE WEBSERVICE 569
 MAPSET option
 CREATE MAPSET command 81
 MAPSET, CREATE command 80
 MAPSETNAME option
 INQUIRE TERMINAL command 495
 SET TERMINAL command 736
 MAXACTIVE option
 INQUIRE TRANCLASS
 command 513
 MAXACTIVE option (*continued*)
 SET TRANCLASS command 751
 MAXDATALEN option
 INQUIRE TCPIP SERVICE
 command 472
 SET TCPIP SERVICE command 727
 MAXIMUM option
 INQUIRE MODENAME
 command 378
 INQUIRE SYSDUMPCODE
 command 439
 INQUIRE TCLASS command 465
 INQUIRE TRANDUMPCODE
 command 516
 SET SYSDUMPCODE command 714
 SET TCLASS command 724
 SET TRANDUMPCODE
 command 754
 MAXITEMLEN option
 INQUIRE TSQNAME command 533
 INQUIRE TSQUEUE command 533
 MAXJVMTCBS option
 INQUIRE DISPATCHER
 command 297
 SET DISPATCHER command 644
 MAXNUMRECS option
 INQUIRE FILE command 341
 SET FILE command 662
 MAXOPENTCBS option
 INQUIRE DISPATCHER
 command 297
 INQUIRE SYSTEM command 447
 SET DISPATCHER command 644
 SET SYSTEM command 718
 MAXPERSIST option
 INQUIRE TCPIP SERVICE
 command 473
 MAXQTIME option
 INQUIRE IPCONN command 348
 MAXREQS option
 INQUIRE AUTOINSTALL
 command 231
 SET AUTOINSTALL command 609
 MAXSOCKETS option
 INQUIRE TCPIP command 466
 SET SYSTEM command 725
 MAXSSLTCBS option
 INQUIRE DISPATCHER
 command 297
 SET DISPATCHER command 644
 MAXTASKS option
 INQUIRE SYSTEM command 447
 SET SYSTEM command 719
 MAXTHRDTCBS option
 INQUIRE DISPATCHER
 command 297
 MAXWINNERS option
 INQUIRE MODENAME
 command 378
 MAXXPTCBS option
 INQUIRE DISPATCHER
 command 297
 SET DISPATCHER command 644
 MEDIATYPE option
 INQUIRE URIMAP command 558
 MEMBER option
 INQUIRE DOCTEMPLATE
 command 304
 INQUIRE TDQUEUE command 482
 MEMBERNAME option
 INQUIRE CONNECTION
 command 264
 MEMLIMIT option
 INQUIRE SYSTEM command 447
 METADATAFILE option
 INQUIRE BUNDLEPART
 command 239
 MINITEMLEN option
 INQUIRE TSQNAME command 533
 INQUIRE TSQUEUE command 533
 MINRUNLEVEL option
 INQUIRE WEBSERVICE 569
 MINRUNRNUM option
 INQUIRE WEBSERVICE 569
 MINRUNVNUM option
 INQUIRE WEBSERVICE 569
 MINUTES option
 INQUIRE REQID command 424
 MODENAME option
 INQUIRE MODENAME
 command 378
 INQUIRE TERMINAL command 495
 SET MODENAME command 692
 MODENAME, INQUIRE command 378
 MODENAME, SET command 691
 MODULE option
 INQUIRE REQUESTMODEL
 command 428
 MONITOR option
 COLLECT STATISTICS command 33
 PERFORM STATISTICS
 command 601
 MONITOR, INQUIRE command 380
 MONITOR, SET command 693
 MQCONN CVDA value
 EXTRACT STATISTICS
 command 210
 MQCONN option
 DISCARD MQCONN command 185
 INQUIRE SYSTEM command 447
 PERFORM STATISTICS
 command 602
 MQCONN, CREATE command 82
 MQCONN, DISCARD command 185
 MQCONN, INQUIRE command 384
 MQCONN, SET command 698
 MQINI, INQUIRE command 388
 MROBATCH option
 INQUIRE DISPATCHER
 command 297
 INQUIRE SYSTEM command 447
 SET DISPATCHER command 644
 SET SYSTEM command 719
 MSRCONTROLST option
 INQUIRE TERMINAL command 495
 MVSIMAGE option
 INQUIRE ASSOCIATION
 command 212
 MVSTCB option
 INQUIRE MVSTCB command 391
 MVSTCB, INQUIRE command 390

N

- name argument, CICS command format 6
- NAMESPACE option
 - INQUIRE BRFACILITY command 234
- NATLANG option
 - INQUIRE TERMINAL command 496
- NATURE option
 - INQUIRE TERMINAL command 496
- NETID option
 - PERFORM ENDAFFINITY command 590
- NETNAME option
 - INQUIRE BRFACILITY command 234
 - INQUIRE CONNECTION command 264
 - INQUIRE PARTNER command 399
 - INQUIRE TERMINAL command 485
 - INQUIRE UOW command 537
 - PERFORM ENDAFFINITY command 590
 - SET NETNAME command 701
- NETNAME, INQUIRE command 392
- NETNAME, SET command 701
- NETUOWID option
 - INQUIRE UOW command 537
 - INQUIRE UOWENQ command 547
 - INQUIRE UOWLINK command 551
- NETWORK option
 - INQUIRE PARTNER command 399
- NEWMAXSOCKET option
 - SET SYSTEM command 725
- NEWMAXTASKS option
 - SET SYSTEM command 719
- NEXTTIME option
 - INQUIRE STATISTICS command 431
- NEXTTIMEHRS option
 - INQUIRE STATISTICS command 431
- NEXTTIMEMINS option
 - INQUIRE STATISTICS command 431
- NEXTTIMESECS option
 - INQUIRE STATISTICS command 432
- NEXTRANSID option
 - INQUIRE TERMINAL command 485
 - SET TERMINAL command 737
- NODE TARGET option
 - COLLECT STATISTICS command 33
- NOHANDLE
 - option 3, 14
- NONCRITICAL value 471
- NOQUEUE option
 - ACQUIRE TERMINAL command 32
- NORESTART option
 - PERFORM SHUTDOWN command 597
- NOSDTRAN option
 - PERFORM SHUTDOWN command 597
- NOSSL value
 - INQUIRE TCPIPSERVICE command 474
- NOTAPPLIC 14
- NOTAPPLIC value 470
- NOTAUTH condition 16
- NOTAUTH condition (*continued*)
 - DISCARD ATOMSERVICE command 170
 - DISCARD URIMAP command 198
 - INQUIRE ATOMSERVICE command 228
 - INQUIRE CAPDATAPRED command 241
 - INQUIRE CAPINFOSRCE command 246
 - INQUIRE CAPOPTPRED command 248
 - INQUIRE CAPTURESPEC command 254
 - INQUIRE EPADAPTER command 322
 - INQUIRE EVENTBINDING command 324
 - INQUIRE EVENTPROCESS command 325
 - INQUIRE HOST command 348
 - INQUIRE JVMSERVER command 373
 - INQUIRE URIMAP command 561
 - INQUIRE WEBSERVICE command 571
 - SET ATOMSERVICE command 608
 - SET EPADAPTER command 660
 - SET EVENTBINDING command 661
 - SET EVENTPROCESS command 662
 - SET HOST command 676
 - SET URIMAP command 764
 - SET WEBSERVICE command 770
 - XMLTRANSFORM command 579, 773
- NOTFND condition
 - DISCARD ATOMSERVICE command 171
 - DISCARD URIMAP command 198
 - INQUIRE ATOMSERVICE command 229
 - INQUIRE CAPDATAPRED command 241
 - INQUIRE CAPINFOSRCE command 246
 - INQUIRE CAPOPTPRED command 248
 - INQUIRE CAPTURESPEC command 254
 - INQUIRE EPADAPTER command 322
 - INQUIRE EVENTBINDING command 324
 - INQUIRE HOST command 348
 - INQUIRE JVMSERVER command 373
 - INQUIRE URIMAP command 561
 - INQUIRE WEBSERVICE command 571
 - SET ATOMSERVICE command 608
 - SET EPADAPTER command 660
 - SET EVENTBINDING command 661
 - SET HOST command 676
 - SET URIMAP command 764
 - SET WEBSERVICE command 770
 - XMLTRANSFORM command 579, 773
- NOTPENDING
 - INQUIRE IPCONN 348
- NOTSUPPORTED value
 - INQUIRE TCPIPSERVICE command 473
- NQNAME option
 - INQUIRE CONNECTION command 265
 - INQUIRE TERMINAL command 485
- null values 14
- NUMCIPHER option
 - INQUIRE CORBASERVER command 275
 - INQUIRE IPCONN command 348
- NUMCIPHERS option
 - INQUIRE TCPIPSERVICE command 473
 - INQUIRE URIMAP command 558
- NUMDATAPRED option
 - INQUIRE CAPTURESPEC command 252
- NUMDSNAMES option
 - INQUIRE LIBRARY command 377
- NUMELEMENTS option
 - INQUIRE MVSTCB command 391
 - INQUIRE STORAGE command 434
- NUMEXITS option
 - INQUIRE EXITPROGRAM command 330
- NUMINFOSRCE option
 - INQUIRE CAPTURESPEC command 252
- NUMITEMS option
 - INQUIRE TDQUEUE command 482
 - INQUIRE TSQNAME command 534
 - INQUIRE TSQUEUE command 534
- NUMOPTPRED option
 - INQUIRE CAPTURESPEC command 252

O

- OAPPLID option
 - INQUIRE WORKREQUEST command 573
- OBFORMATST option
 - INQUIRE TERMINAL 737
 - INQUIRE TERMINAL command 485
- OBJECT option
 - INQUIRE DSNNAME command 309
 - INQUIRE FILE command 341
- OBOPERIDST option
 - INQUIRE TERMINAL command 485
- ODAPPLID option
 - INQUIRE ASSOCIATION command 212
- ODCLNTIPADDR option
 - INQUIRE ASSOCIATION command 212
- ODCLNTPORT option
 - INQUIRE ASSOCIATION command 212
- ODFACILNAME option
 - INQUIRE ASSOCIATION command 212

ODFACILTYPE option
 INQUIRE ASSOCIATION
 command 212

ODIPFAMILY option
 INQUIRE ASSOCIATION
 command 212

ODLUNAME option
 INQUIRE ASSOCIATION
 command 212

ODSTARTTIME option
 INQUIRE ASSOCIATION
 command 212

ODTASKID option
 INQUIRE ASSOCIATION
 command 212

ODTRANSID option
 INQUIRE ASSOCIATION
 command 212

ODUSERID option
 INQUIRE ASSOCIATION
 command 212

OLDCACHES option
 INQUIRE CLASSCACHE
 command 257

OPEN value
 INQUIRE TCPIP command 466
 INQUIRE TCPIPSERVICE
 command 473
 SET TCPIP command 725
 SET TCPIPSERVICE command 727

OPENING value
 INQUIRE TCPIPSERVICE
 command 473

OPENSTATUS option
 INQUIRE DUMPDS command 313
 INQUIRE FILE command 342
 INQUIRE IRC command 359
 INQUIRE RRMS command 429
 INQUIRE TCPIP command 466
 INQUIRE TDQUEUE command 482
 INQUIRE VTAM command 563
 SET DUMPDS command 657
 SET FILE command 662
 SET IRC command 682
 SET TCPIP command 725
 SET TCPIPSERVICE command 727
 SET TDQUEUE command 731
 SET VTAM command 766

OPERATION option
 INQUIRE REQUESTMODEL
 command 428

OPERATOR option
 INQUIRE CAPDATAPRED
 command 242
 INQUIRE CAPOPTPRED
 command 247

OPERID option
 INQUIRE TERMINAL command 485
 SET TERMINAL command 737

OPID
 option of DSNCRCT macro 276, 637

OPREL option
 INQUIRE SYSTEM command 447

OPSYS option
 INQUIRE SYSTEM command 447

OPTIONNAME option
 INQUIRE CAPOPTPRED
 command 248

OSGIBUNDLE, INQUIRE command 393

OSGISERVICE, INQUIRE command 397

OSLEVEL
 CEMT INQUIRE SYSTEM 447

OTASK option
 INQUIRE WORKREQUEST
 command 573

OTRANSID option
 INQUIRE WORKREQUEST
 command 573

OTSTID option
 INQUIRE UOW command 537

OTSTIMEOUT option
 INQUIRE TRANSACTION
 command 522, 757

OUTLINEST option
 INQUIRE TERMINAL command 485

OUTPRIVACY option
 INQUIRE CORBASERVER
 command 275

P

PAGEHT option
 INQUIRE TERMINAL command 485

PAGESTATUS option
 INQUIRE TERMINAL command 485
 SET TERMINAL command 737

PAGEWD option
 INQUIRE TERMINAL command 485

PARTCLASS option
 INQUIRE BUNDLEPART
 command 239

PARTCOUNT option
 INQUIRE BUNDLE command 238

PARTIAL option
 RESYNC command 607

PARTITIONSET option
 CREATE PARTITIONSET
 command 84

PARTITIONSET, CREATE command 83

PARTITIONSST option
 INQUIRE TERMINAL command 485

PARTNER option
 CREATE PARTNER command 86
 DISCARD PARTNER command 186
 INQUIRE PARTNER command 399

PARTNER, CREATE command 85

PARTNER, DISCARD command 186

PARTNER, INQUIRE command 398

PARTTYPE option
 INQUIRE BUNDLEPART
 command 239

PATH option
 INQUIRE URIMAP command 558

PDSMEMBER option
 INQUIRE DOCTEMPLATE
 command 305

PENDING
 INQUIRE IPCONN 348

PENDSTATUS option
 INQUIRE CONNECTION
 command 265
 INQUIRE IPCONN command 348

PENDSTATUS option (*continued*)
 SET CONNECTION command 616
 SET IPCONN command 677

PERFCLASS option
 INQUIRE MONITOR command 383
 SET MONITOR command 696

PERFORM CLASSCACHE
 command 580
 conditions 582

PERFORM commands
 CLASSCACHE 580
 DELETSHIPPED 585
 DUMP 587
 ENDAFFINITY 590
 JVMPOOL 591
 RESETTIME 594
 SECURITY REBUILD 595
 SHUTDOWN 596
 STATISTICS RECORD 598

PERFORM CORBASERVER 582

PERFORM CORBASERVER command
 conditions 584

PERFORM DELETSHIPPED
 command 585

PERFORM DJAR 586

PERFORM DJAR command
 conditions 587

PERFORM DUMP command 587
 conditions 587

PERFORM ENDAFFINITY
 command 590
 conditions 590

PERFORM JVMPOOL command 591
 conditions 592

PERFORM PIPELINE 593

PERFORM PIPELINE command
 conditions 594

PERFORM RESETTIME command 594
 conditions 594

PERFORM SECURITY REBUILD
 command 595
 conditions 595

PERFORM SHUTDOWN command 596
 conditions 598

PERFORM STATISTICS RECORD
 command 598
 conditions 603

PGMINTERFACE option
 INQUIRE WEBSERVICE
 command 569

PHASINGOUT option
 INQUIRE CLASSCACHE
 command 257
 INQUIRE JVMPOOL command 368

PHASINGOUTST option
 INQUIRE JVM command 366

PIPELINE CVDA value
 EXTRACT STATISTICS
 command 210

PIPELINE option
 CREATE PIPELINE command 88
 INQUIRE URIMAP command 558
 INQUIRE WEBSERVICE
 command 569
 PERFORM PIPELINE command 593
 PERFORM STATISTICS
 command 602

PIPELINE, CREATE command 87
 PIPELINE, DISCARD command 187
 PIPELINE, INQUIRE command 400
 PIPELINE, PERFORM 593
 PIPELINE, SET command 702
 PL/I language
 argument values 12
 LENGTH option default 11
 PLT option
 PERFORM SHUTDOWN
 command 597
 pointer arguments 8
 pointer-ref argument, CICS command
 format 6
 pointer-value argument, CICS command
 format 6
 POOL option
 COLLECT STATISTICS command 33
 POOL TARGET option
 COLLECT STATISTICS command 33
 POOLNAME option
 INQUIRE TSQNAME command 534
 INQUIRE TSQUEUE command 534
 PORT option
 INQUIRE IPCONN command 348
 INQUIRE TCPIP SERVICE
 command 473
 INQUIRE URIMAP command 559
 PRIMPREP option
 INQUIRE CAPTURESPEC
 command 252
 PRIMPREP DOP option
 INQUIRE CAPTURESPEC
 command 253
 PRIMPREP TYPE option
 INQUIRE CAPTURESPEC
 command 253
 PRINTADAPTST option
 INQUIRE TERMINAL command 485
 PRINTCONTROL option
 INQUIRE TDQUEUE command 482
 PRINTER option
 INQUIRE TERMINAL command 485
 SET TERMINAL command 737
 PRIORITY option
 INQUIRE TASK command 457
 INQUIRE TRANSACTION
 command 522
 SET TASK command 722
 SET TRANSACTION command 757
 PRIVACY option
 INQUIRE TCPIP SERVICE
 command 473
 PROCESS option
 INQUIRE TASK command 457
 PROCESSTYPE option
 CREATE PROCESSTYPE
 command 90
 INQUIRE PROCESSTYPE
 command 407, 704
 INQUIRE TASK command 457
 PROCESSTYPE, CREATE command 89
 PROCESSTYPE, DISCARD
 command 188
 PROCESSTYPE, INQUIRE
 command 405
 PROCESSTYPE, SET command 703
 PROFILE option
 CREATE PROFILE command 92
 DISCARD PROFILE command 189
 INQUIRE JVM command 366
 INQUIRE PARTNER command 399
 INQUIRE PROFILE command 409
 INQUIRE TASK command 457
 INQUIRE TRANSACTION
 command 522
 PROFILE, DISCARD command 188
 PROFILE, INQUIRE command 408
 PROFILEDIR
 INQUIRE JVMPOOL command 368
 PROGAUTO option
 COLLECT STATISTICS command 33
 PERFORM STATISTICS
 command 602
 PROGAUTOCTLG option
 INQUIRE SYSTEM command 447
 SET SYSTEM command 719
 PROGAUTOEXIT option
 INQUIRE SYSTEM command 448
 SET SYSTEM command 719
 PROGAUTOINST option
 INQUIRE SYSTEM command 448
 SET SYSTEM command 720
 PROGRAM option
 COLLECT STATISTICS command 33
 CREATE PROGRAM command 93
 DISABLE PROGRAM command 168
 DISCARD PROGRAM command 190
 ENABLE PROGRAM command 202
 EXTRACT EXIT command 206
 INQUIRE ASSOCIATION
 command 212
 INQUIRE AUTOINSTALL
 command 231
 INQUIRE DOCTEMPLATE
 command 305
 INQUIRE PROGRAM command 410
 INQUIRE TASK command 458
 INQUIRE TRANSACTION
 command 522
 INQUIRE URIMAP command 559
 INQUIRE WEBSERVICE
 command 569
 PERFORM STATISTICS
 command 602
 SET AUTOINSTALL command 610
 SET PROGRAM command 707
 PROGRAM, CREATE command 93
 PROGRAM, DISABLE command 167
 PROGRAM, DISCARD command 189
 PROGRAM, ENABLE command 199
 PROGRAM, INQUIRE command 410
 PROGRAM, SET command 704
 PROGRAMDEF option
 PERFORM STATISTICS
 command 602
 PROGSYMBOLST option
 INQUIRE TERMINAL command 485
 PROGTYPE option
 INQUIRE PROGRAM command 410
 PROTOCOL option
 INQUIRE CONNECTION
 command 265
 PROTOCOL option (*continued*)
 INQUIRE TCPIP SERVICE
 command 473
 INQUIRE UOWLINK command 551
 PRTCOPYST option
 INQUIRE TERMINAL command 485
 SET TERMINAL command 738
 PRTYAGING option
 INQUIRE DISPATCHER
 command 297
 INQUIRE SYSTEM command 448
 SET DISPATCHER command 645
 SET SYSTEM command 720
 PSDINTERVAL option
 INQUIRE VTAM command 563
 SET VTAM 766
 PSDINTHRS option
 INQUIRE VTAM command 564
 SET the Communications Server 767
 PSDINTMINS option
 INQUIRE VTAM command 564
 SET VTAM 767
 PSDINTSECS option
 INQUIRE VTAM command 564
 SET VTAM 767
 PSTYPE option
 INQUIRE VTAM command 564
 PURGEABILITY option
 INQUIRE TASK command 458
 INQUIRE TRANSACTION
 command 523
 SET TRANSACTION command 757
 PURGEABLE option
 DISABLE PROGRAM command 168
 PURGEABLEST option
 INQUIRE EXITPROGRAM
 command 330
 PURGETHRESH option
 INQUIRE TRANCLASS
 command 513
 SET TRANCLASS command 752
 PURGETYPE option
 SET CONNECTION command 616
 SET IPCONN command 678
 SET TASK command 722
 SET TERMINAL command 738
 SET WORKREQUEST command 771

Q

QALL option
 ACQUIRE TERMINAL command 32
 QNOTETAB option
 ACQUIRE TERMINAL command 32
 QSESSLIM option
 ACQUIRE TERMINAL command 32
 QUALIFIER option
 INQUIRE EXITPROGRAM
 command 330
 RESYNC command 607
 QUALLEN option
 INQUIRE UOWENQ command 547
 QUERY SECURITY command 19
 QUERYST option
 INQUIRE TERMINAL command 485
 QUEUE option
 INQUIRE REQID command 424

QUEUED option
 INQUIRE TRANCLASS
 command 514
 QUEUELIMIT option
 INQUIRE IPCONN command 348
 QUIESCSTATE option
 INQUIRE DSNAME 309

R

RACF (resource access control facility) 2
 RANKING option
 INQUIRE LIBRARY command 377
 SET LIBRARY command 690
 RBATYPE option
 INQUIRE FILE command 342
 RDSASIZE option
 INQUIRE SYSTEM command 448
 READ option
 INQUIRE FILE command 342
 SET FILE command 662
 READINTEG option
 INQUIRE FILE command 343
 SET FILE command 662
 REALM option
 INQUIRE ASSOCIATION
 command 212
 INQUIRE ASSOCIATION LIST
 command 222
 INQUIRE TCPIP SERVICE
 command 474
 REALMLen option
 INQUIRE ASSOCIATION LIST
 command 222
 RECEIVECOUNT option
 INQUIRE CONNECTION
 command 266
 INQUIRE IPCONN command 348
 RECORDFORMAT option
 INQUIRE FILE command 343
 INQUIRE TDQUEUE command 483
 RECORDING option
 INQUIRE STATISTICS command 432
 SET STATISTICS command 711
 RECORDLENGTH option
 INQUIRE TDQUEUE command 483
 RECORDNOW option
 SET STATISTICS command 711
 RECORDSIZE option
 INQUIRE FILE command 343
 SET FILE command 662
 RECOVERY option
 PERFORM STATISTICS
 command 602
 RECOVSTATUS option
 INQUIRE CONNECTION
 command 266
 INQUIRE DSNAME command 310
 INQUIRE FILE command 344
 INQUIRE IPCONN command 348
 INQUIRE TDQUEUE command 483
 INQUIRE TSQNAME command 534
 INQUIRE TSQUEUE command 534
 SET CONNECTION command 618
 SET IPCONN command 679
 REDIRECTTYPE option
 INQUIRE URIMAP command 559

REDIRECTTYPE option (*continued*)
 SET URIMAP command 764
 REENTPROTECT option
 INQUIRE SYSTEM command 448
 REGERROR value 471
 REGISTERED value 471
 RELATION option
 INQUIRE UOWENQ command 547
 RELEASE option
 INQUIRE SYSTEM command 448
 RELOAD option
 PERFORM CLASSCACHE
 command 581
 relocatable expression 13
 RELREQ option
 ACQUIRE TERMINAL command 32
 RELREQST option
 INQUIRE TERMINAL command 485
 SET TERMINAL command 739
 RELTYPE option
 INQUIRE FILE command 344
 remote definition, not retrievable or
 updateable 1
 REMOTENAME option
 INQUIRE CONNECTION
 command 266
 INQUIRE FILE command 344
 INQUIRE PROGRAM command 410
 INQUIRE TASK command 458
 INQUIRE TDQUEUE command 483
 INQUIRE TERMINAL command 485
 INQUIRE TRANSACTION
 command 523
 REMOTESYSNET option
 INQUIRE BR FACILITY
 command 234
 INQUIRE CONNECTION
 command 266
 INQUIRE TERMINAL command 485
 REMOTESYSTEM option
 INQUIRE BR FACILITY
 command 234
 INQUIRE CONNECTION
 command 266
 INQUIRE FILE command 344
 INQUIRE PROGRAM command 410
 INQUIRE TASK command 458
 INQUIRE TDQUEUE command 483
 INQUIRE TERMINAL command 485
 INQUIRE TRANSACTION
 command 523
 REMOTETABLE option
 INQUIRE FILE command 344
 REMOVE command
 CSD 154
 removing 649
 RENAME command
 CSD 156
 REQID option
 INQUIRE REQID command 424
 REQID, INQUIRE command 421
 REQTYPE option
 INQUIRE REQID command 424
 REQUESTID option
 INQUIRE WORKREQUEST
 command 573

REQUESTMODEL option
 CREATE REQUESTMODEL
 command 97
 PERFORM STATISTICS
 command 602
 REQUESTMODEL, CREATE
 command 95
 REQUESTMODEL, DISCARD
 command 190
 REQUESTMODEL, INQUIRE
 command 426
 REQUIRED value
 INQUIRE TCPIP SERVICE
 command 473
 RESCOUNT option
 INQUIRE PROGRAM command 410
 RESETNOW option
 PERFORM STATISTICS
 command 602
 SET STATISTICS command 711
 RESETTIME, PERFORM command 594
 RESLEN option
 INQUIRE UOWENQ command 548
 RESNAME option
 INQUIRE TASK command 458
 resource access control facility (RACF) 2
 RESOURCE option
 INQUIRE UOWENQ command 548
 resource security checking 17
 RESOURCENAME option
 INQUIRE ATOMSERVICE
 command 228
 resources
 class (ESM) 16
 RESOURCETYPE option
 INQUIRE ATOMSERVICE
 command 228
 RESP and RESP2 options
 values returned 16
 RESP options 15
 RESP2
 option 14
 RESP2 options 15
 RESP2 values
 EXEC CICS CREATE 837
 Response Codes
 of EXEC CICS commands 825
 RESRCECLASS option
 INQUIRE MONITOR command 383
 SET MONITOR command 696
 RESSEC option
 INQUIRE TASK command 458
 INQUIRE TRANSACTION
 command 523
 RESTYPE option
 EXTRACT STATISTICS
 command 209, 210
 RESYNC command
 ENTRYNAME 607
 IDLIST 607
 IDLISTLENGTH 607
 PARTIAL 607
 QUALIFIER 607
 RESYNC ENTRYNAME command 605
 conditions 608
 RESYNCSSTATUS option
 INQUIRE UOWLINK command 552

- RETLOCKS option
 - INQUIRE DSNNAME 311
 - REUSEST option
 - INQUIRE CLASSCACHE command 256
 - INQUIRE JVM command 366, 369
 - REWIND option
 - INQUIRE TDQUEUE command 484
 - RLSACCESS option
 - INQUIRE FILE command 345
 - SET FILE command 662
 - RMIQFY option
 - INQUIRE UOWLINK command 552
 - RMIST option
 - INQUIRE MONITOR command 383
 - ROLE option
 - INQUIRE UOWLINK command 552
 - ROUTESTATUS option
 - INQUIRE TRANSACTION command 523
 - ROUTING option
 - INQUIRE TASK command 459
 - INQUIRE TRANSACTION command 523
 - RRMS, INQUIRE command 429
 - RTERMID option
 - INQUIRE REQID command 424
 - RTIMEOUT option
 - INQUIRE TASK command 459
 - INQUIRE TRANSACTION command 524
 - RTRANSID option
 - INQUIRE REQID command 424
 - rules for browsing 22
 - RUNAWAY option
 - INQUIRE DISPATCHER command 298
 - INQUIRE SYSTEM command 449
 - INQUIRE TASK command 459
 - INQUIRE TRANSACTION command 524
 - SET DISPATCHER command 645
 - SET SYSTEM command 720
 - SET TRANSACTION command 757
 - RUNAWAYTYPE option
 - INQUIRE TRANSACTION command 524
 - SET TRANSACTION command 757
 - RUNNING option
 - INQUIRE TASK LIST command 464
 - RUNSTATUS option
 - INQUIRE TASK command 459
 - RUNTIME option
 - INQUIRE PROGRAM command 410
 - SET PROGRAM command 707
- S**
- SCANDELAY option
 - INQUIRE DISPATCHER command 298
 - INQUIRE SYSTEM command 449
 - SET DISPATCHER command 645
 - SET SYSTEM command 720
 - SCHEMALEVEL option
 - INQUIRE EVENTPROCESS 325
 - SCHEME option
 - INQUIRE URIMAP command 559
 - SCRNHT option
 - INQUIRE TERMINAL command 485
 - SCRNSIZE option
 - INQUIRE TASK command 459
 - INQUIRE TRANSACTION command 524
 - SCRNWD option
 - INQUIRE TERMINAL command 485
 - SDSASIZE option
 - INQUIRE SYSTEM command 449
 - SDTRAN option
 - INQUIRE SYSTEM command 449
 - PERFORM SHUTDOWN command 597
 - SEARCHPOS option
 - INQUIRE LIBRARY command 377
 - SECONDS option
 - INQUIRE REQID command 424
 - security
 - command 16
 - NOTAUTH condition 16
 - QUERY SECURITY command 19
 - resource security checking 16
 - security checking by ESM 16
 - security check failures 16
 - security checking
 - command 16
 - resource 17
 - surrogate 16
 - transaction 16
 - SECURITY option
 - INQUIRE TERMINAL command 485
 - SECURITY REBUILD, PERFORM command 595
 - SECURITYMGR option
 - INQUIRE SYSTEM command 449
 - SECURITYNAME option
 - INQUIRE IPCONN command 348
 - SENDCOUNT option
 - INQUIRE CONNECTION command 267
 - INQUIRE IPCONN command 348
 - SERVERIPADDR option
 - INQUIRE ASSOCIATION command 212
 - SERVERPORT option
 - INQUIRE ASSOCIATION command 212
 - SERVSTATUS option
 - INQUIRE CONNECTION command 267
 - INQUIRE IPCONN command 348
 - INQUIRE TERMINAL command 485
 - SET CONNECTION command 618
 - SET IPCONN command 679
 - SET TERMINAL command 739
 - SESSBEANTIME option
 - INQUIRE CORBASERVER command 275
 - SET CORBASERVER command 623
 - SESSIONS option
 - CREATE SESSIONS command 99
 - SESSIONTYPE option
 - INQUIRE TERMINAL command 485
 - SET ATOMSERVICE command 608
 - SET ATOMSERVICE command
 - (continued)
 - conditions 608
 - SET AUTOINSTALL command 609
 - conditions 610
 - SET BRFACILITY command 610
 - conditions 611
 - SET BUNDLE command 611
 - conditions 612
 - SET CLASSCACHE command 612
 - conditions 613
 - SET commands 23
 - AUTOINSTALL 609
 - BRFACILITY 610
 - BUNDLE 611
 - CLASSCACHE 612
 - CONNECTION 614
 - CORBASERVER 622
 - DB2CONN 624
 - DB2ENTRY 634
 - DB2TRAN 640
 - DELETSHIPPED 641
 - DISPATCHER 643
 - DSNAME 647
 - DUMPDS 656
 - EPADAPTER 659
 - EVENTBINDING 660
 - EVENTPROCESS 661
 - FILE 662
 - IPCONN 676
 - IRC 681
 - JOURNALNUM 685
 - LIBRARY 689
 - MODENAME 691
 - MONITOR 693
 - MQCONN 698
 - NETNAME 701
 - PIPELINE 702
 - PROGRAM 704
 - STATISTICS 709
 - SYSDUMPCODE 712
 - SYSTEM 715
 - TASK 721
 - TCLASS 723
 - TCPIP 724
 - TCPIPSERVICE 726
 - TDQUEUE 728
 - TERMINAL 733
 - TRACEDEST 742
 - TRACEFLAG 746
 - TRACETYPE 748
 - TRANCLASS 751
 - TRANDUMPCODE 752
 - TRANSACTION 756
 - UOWLINK 762
 - VOLUME 765
 - SET commandsz/OS Communications
 - Server connection for CICS VTAM 765
 - SET CONNECTION command 614
 - conditions 620
 - SET CORBASERVER command 622
 - conditions 624
 - SET DB2CONN command 624
 - conditions 624
 - SET DB2ENTRY command 634
 - conditions 639

SET DB2TRAN command 640
 conditions 640
 SET DELETSHIPED command 641
 conditions 643
 SET DISPATCHER command 643
 conditions 645
 SET DOCTEMPLATE command 646
 conditions 647
 SET DSNNAME command 647
 conditions 652
 SET DUMPDS command 656
 conditions 658
 SET ENQMODEL command 658
 conditions 659
 SET EPADAPTER command 659
 conditions 660
 SET EVENTBINDING command 660
 conditions 661
 SET EVENTPROCESS command 661
 conditions 662
 SET FILE command 662
 conditions 662
 SET HOST command 675
 conditions 676
 SET IPCONN command 676
 conditions 680
 SET IRC command 681
 conditions 682
 SET JOURNALNAME command 683
 conditions 684
 SET JOURNALNUM command 685
 SET JVMPOOL command 685
 conditions 686
 SET JVMSEVER command 686
 conditions 688
 SET LIBRARY command 689
 SET MODENAME command 691
 conditions 692
 SET MONITOR command 693
 conditions 697
 SET MQCONN command 698
 conditions 698
 SET NETNAME command 701
 conditions 701
 SET option
 COLLECT STATISTICS command 33
 EXTRACT STATISTICS
 command 210
 INQUIRE ASSOCIATION LIST
 command 222
 INQUIRE MVSTCB command 391
 INQUIRE REQID command 424
 INQUIRE TASK LIST command 464
 SET PIPELINE command 702
 conditions 702
 SET PROCESSTYPE command 703
 SET PROGRAM command 704
 conditions 690, 708
 SET STATISTICS command 709
 conditions 711
 SET SYSDUMPCODE command 712
 conditions 715
 SET SYSTEM command 715
 conditions 720
 SET TASK command 721
 conditions 723
 SET TCLASS command 723
 conditions 724
 SET TCPIP command 724
 conditions 725
 SET TCPIPSERVICE command 726
 conditions 727
 SET TDQUEUE command 728
 conditions 731
 SET TEMPSTORAGE command 732
 conditions 733
 SET TERMINAL command 733
 conditions 741
 SET TRACEDEST command 742
 conditions 745
 SET TRACEFLAG command 746
 conditions 748
 SET TRACETYPE command 748
 conditions 750
 SET TRANCLASS command 751
 conditions 752
 SET TRANDUMPCODE command 752
 conditions 755
 SET TRANSACTION command 756
 conditions 759
 SET TSQNAME command 759
 conditions 760
 SET TSQUEUE command 759
 conditions 760
 SET UOW command 761
 conditions 761
 SET UOWLINK command 762
 conditions 762
 SET URIMAP command 763
 conditions 764
 SET VOLUME command 765
 conditions 765
 SET VTAM command 765
 conditions 767
 SET WEBSERVICE command 769
 conditions 770
 SET WORKREQUEST command 770
 conditions 771
 SET XMLTRANSFORM command 772
 SETTRANSID option
 INQUIRE TASK LIST command 464
 SHARESTATUS option
 INQUIRE PROGRAM command 410
 SET PROGRAM command 707
 SHELF option
 INQUIRE CORBASERVER
 command 275
 SHUTDOWN option
 DISABLE PROGRAM command 168
 ENABLE PROGRAM command 202
 INQUIRE TRANSACTION
 command 524
 SET TRANSACTION command 758
 SHUTDOWN, PERFORM command 596
 SHUTDOWNST option
 INQUIRE EXITPROGRAM
 command 331
 SHUTOPTION option
 INQUIRE SYSDUMPCODE
 command 439
 INQUIRE TRANDUMPCODE
 command 516
 SET SYSDUMPCODE command 714
 SHUTOPTION option (continued)
 SET TRANDUMPCODE
 command 754
 SHUTSTATUS option
 INQUIRE SYSTEM command 449
 SIGNID option of DSNCRCT macro 276,
 288
 SIGNONSTATUS option
 INQUIRE TERMINAL command 485
 SINGLESTATUS option
 INQUIRE TRACEFLAG
 command 507
 SET TRACEFLAG command 746
 SOCKETCLOSE option
 INQUIRE TCPIPSERVICE
 command 474
 INQUIRE URIMAP command 559
 SOCKPOOLSIZ option
 INQUIRE URIMAP command 559
 SOSABOVEBAR option
 INQUIRE SYSTEM command 450
 SOSABOVELINE option
 INQUIRE SYSTEM command 450
 SOSBELOWLINE option
 INQUIRE SYSTEM command 450
 SOSIST option
 INQUIRE TERMINAL command 485
 SOSSTATUS option
 INQUIRE SYSTEM command 450
 SPI option
 DISABLE PROGRAM command 168
 ENABLE PROGRAM command 202
 SPIST option
 INQUIRE EXITPROGRAM
 command 331
 SRVRIPFAMILY option
 INQUIRE ASSOCIATION
 command 212
 SSL option
 PERFORM CORBASERVER
 command 583
 PERFORM DJAR command 586
 SSL value
 INQUIRE TCPIPSERVICE
 command 474
 SSLCACHE option
 INQUIRE TCPIP command 466
 SSLTYPE option
 INQUIRE IPCONN command 348
 INQUIRE TCPIPSERVICE
 command 474
 SSLUNAUTH option
 INQUIRE CORBASERVER
 command 275
 STACK option
 INQUIRE WORKREQUEST
 command 573
 START option
 ENABLE PROGRAM command 202
 PERFORM CLASSCACHE
 command 581
 STARTBRGROUP command
 CSD 159
 STARTBRLIST command
 CSD 160
 STARTBRRSRCE command
 CSD 161

STARTCODE option
 INQUIRE TASK command 459

starting a browse 21

STARTSTATUS option
 INQUIRE EXITPROGRAM
 command 331

STARTTIME option
 INQUIRE ASSOCIATION
 command 212
 INQUIRE CLASSCACHE
 command 257

STARTUP option
 INQUIRE SYSTEM command 451

STARTUPDATE option
 INQUIRE SYSTEM command 451

STATE option
 INQUIRE DJAR command 301
 INQUIRE UOWENQ command 548
 INQUIRE WEBSERVICE
 command 569

STATISTICS RECORD, PERFORM
 command 598

STATISTICS, COLLECT command 33

STATISTICS, EXTRACT command 207

STATISTICS, INQUIRE command 430

STATISTICS, SET command 709

STATS option
 COLLECT STATISTICS command 33
 PERFORM STATISTICS
 command 602

STATUS option
 INQUIRE CLASSCACHE
 command 258
 INQUIRE ENQMODEL
 command 316
 INQUIRE JVMPOOL command 368
 INQUIRE MONITOR command 383
 INQUIRE PROCESSTYPE
 command 407, 704
 INQUIRE PROGRAM command 410
 INQUIRE TCPIP SERVICE
 command 473
 INQUIRE TRANSACTION
 command 524
 SET JVMPOOL command 686
 SET MONITOR command 697
 SET PROGRAM command 707
 SET TRANSACTION command 758

STOP option
 DISABLE PROGRAM command 168

STORAGE option
 COLLECT STATISTICS command 33
 PERFORM STATISTICS
 command 602

STORAGE, INQUIRE command 432

STORAGECLEAR option
 INQUIRE TASK command 460
 INQUIRE TRANSACTION
 command 524

STOREPROTECT option
 INQUIRE SYSTEM command 451

STREAMNAME option
 PERFORM STATISTICS
 command 602

STREAMNAME, INQUIRE
 command 435

STRINGS option
 INQUIRE FILE command 345
 SET FILE command 662

STRUCTNAME option
 INQUIRE CAPDATAPRED
 command 243
 INQUIRE CAPINFOSRCE
 command 245

SUBPOOL option
 COLLECT STATISTICS command 33
 INQUIRE SUBPOOL command 437
 INQUIRE TEMPSTORAGE
 command 485

SUBPOOL, INQUIRE command 436

SUBPOOLLIST option
 INQUIRE MVSTCB command 391

SUBRESTYPE CVDA value
 EXTRACT STATISTICS
 command 210

SUBTASKS option
 INQUIRE DISPATCHER
 command 298

SUPPORTED value
 INQUIRE TCPIP SERVICE
 command 473

surrogate security checking 16

SUSPENDED option
 INQUIRE TASK LIST command 464

SUSPENDTIME option
 INQUIRE TASK command 460

SUSPENDTYPE option
 INQUIRE TASK command 460

SUSPENDVALUE option
 INQUIRE TASK command 460

SWITCHACTION option
 SET TRACEDEST command 744

SWITCHSTATUS option
 INQUIRE DUMPDS command 313
 INQUIRE TRACEDEST
 command 506
 SET DUMPDS command 657
 SET TRACEDEST command 744

SYNCPPOINTST option
 INQUIRE MONITOR command 384
 SET MONITOR command 697

syntax notation 3

SYSDUMP option
 PERFORM STATISTICS
 command 602

SYSDUMPCODE option
 COLLECT STATISTICS command 33
 SET SYSDUMPCODE command 714

SYSDUMPCODE, INQUIRE
 command 437

SYSDUMPCODE, SET command 712

SYSDUMPING option
 INQUIRE SYSDUMPCODE
 command 439
 INQUIRE TRANDUMPCODE
 command 516
 SET SYSDUMPCODE command 714
 SET TRANDUMPCODE
 command 754

SYSEIB
 option 3

SYSID option
 INQUIRE TSQNAME command 534

SYSID option (*continued*)
 INQUIRE TSQUEUE command 534
 INQUIRE UOW command 537
 INQUIRE UOWLINK command 552

SYSOUTCLASS option
 INQUIRE TDQUEUE command 484

SYSPLEX value
 INQUIRE TCPIP command 466

system connections 172, 187, 193, 198,
 258, 348

system programming commands 1
 inquiry 19

SYSTEM, INQUIRE command 440

SYSTEM, SET command 715

systemname argument, CICS command
 format 6

SYSTEMSTATUS option
 INQUIRE TRACEFLAG
 command 508
 SET TRACEFLAG command 747

T

TABLE option
 INQUIRE FILE command 345
 SET FILE command 662

TABLEMGR option
 COLLECT STATISTICS command 33
 PERFORM STATISTICS
 command 602

TABlename option
 INQUIRE FILE command 345
 SET FILE command 662

TABLESIZE option
 INQUIRE TRACEDEST
 command 507
 SET TRACEDEST command 745

TAKEOVER option
 PERFORM SHUTDOWN
 command 597

TALength option
 ENABLE PROGRAM command 203
 INQUIRE EXITPROGRAM
 command 331

TARGET NODE option
 COLLECT STATISTICS command 33

TARGETCOUNT option
 INQUIRE BUNDLE command 238

TARGETSYS option
 INQUIRE WORKREQUEST
 command 573

TASK LIST, INQUIRE command 463

TASK option
 INQUIRE EXCI command 326
 INQUIRE JVM command 366
 INQUIRE STORAGE command 434
 INQUIRE TASK command 461
 INQUIRE UOW command 537
 INQUIRE WORKREQUEST
 command 574
 SET TASK command 722

task-related user exits, restart
 resynchronization 605

TASK, INQUIRE command 452

TASK, SET command 721

TASKDATAKEY option
 INQUIRE TASK command 461

TASKDATAKEY option (*continued*)
 INQUIRE TRANSACTION
 command 525

TASKDATALOC option
 INQUIRE TASK command 461
 INQUIRE TRANSACTION
 command 525

TASKID option
 INQUIRE BRFACILITY
 command 234
 INQUIRE TERMINAL command 485
 INQUIRE UOWENQ command 548

TASKSTART option
 DISABLE PROGRAM command 169
 ENABLE PROGRAM command 203

TASKSTARTST option
 INQUIRE EXITPROGRAM
 command 331

TASKSUBPOOL option
 COLLECT STATISTICS command 33

TCAMCONTROL option
 INQUIRE TERMINAL command 485
 SET TERMINAL command 739

TCB option
 INQUIRE TASK command 461

TCEXITSTATUS option
 INQUIRE TRACEFLAG
 command 508
 SET TRACEFLAG command 747

TCLASS option
 COLLECT STATISTICS command 33
 INQUIRE TASK command 462
 INQUIRE TRANSACTION
 command 525
 PERFORM STATISTICS
 command 602
 SET TRANSACTION command 758

TCLASS, INQUIRE command 464

TCLASS, SET command 723

TCPIP option
 COLLECT STATISTICS command 33
 PERFORM STATISTICS
 command 602

TCPIP, INQUIRE command 466

TCPIP, SET command 724

TCPIPJOB option
 INQUIRE ASSOCIATION
 command 212

TCPIPSERVICE option
 COLLECT STATISTICS command 33
 CREATE TCPIPSERVICE
 command 102
 INQUIRE ASSOCIATION
 command 212
 INQUIRE IPCONN command 348
 INQUIRE TCPIPSERVICE
 command 474
 INQUIRE URIMAP command 347,
 560
 PERFORM STATISTICS
 command 602

TCPIPSERVICE, DISCARD
 command 191

TCPIPSERVICE, INQUIRE
 command 467

TCPIPSERVICE, SET command 726

TCPIPZONE option
 INQUIRE ASSOCIATION
 command 212

TDQ option
 INQUIRE DOCTEMPLATE
 command 305

TDQUEUE option
 COLLECT STATISTICS command 33
 CREATE TDQUEUE command 106
 DISCARD TDQUEUE command 192
 INQUIRE DOCTEMPLATE
 command 305
 INQUIRE TDQUEUE command 484
 PERFORM STATISTICS
 command 602
 SET TDQUEUE command 731

TDQUEUE, DISCARD command 192

TDQUEUE, INQUIRE command 475

TDQUEUE, SET command 728

TEMPLATENAME option
 INQUIRE DOCTEMPLATE
 command 305
 INQUIRE URIMAP command 560

TEMPLATETYPE option
 INQUIRE DOCTEMPLATE
 command 305

TEMPSTORAGE, INQUIRE
 command 485

TERM option
 DSNCRCT macro 276, 288, 624, 636

TERMINAL option
 INQUIRE BRFACILITY
 command 234
 INQUIRE REQID command 424
 INQUIRE UOW command 537

TERMINAL option
 ACQUIRE TERMINAL command 32
 COLLECT STATISTICS command 33
 CREATE TERMINAL command 108
 INQUIRE TERMINAL command 485

PERFORM STATISTICS
 command 603
 SET TERMINAL command 739

TERMINAL, ACQUIRE command 31

TERMINAL, CREATE command 106

TERMINAL, DISCARD command 193

TERMINAL, INQUIRE command 485

TERMINAL, SET command 733

TERMINATE option
 PERFORM CLASSCACHE
 command 581
 PERFORM JVMPOOL command 591
 SET JVMPOOL command 686

TERMMODEL option
 INQUIRE TERMINAL command 485

TERMPRIORITY option
 INQUIRE TERMINAL command 485
 SET TERMINAL command 739

TERMSTATUS option
 INQUIRE BRFACILITY
 command 234
 INQUIRE TERMINAL command 485
 SET BRFACILITY command 611
 SET TERMINAL command 739

TEXTKYBDST option
 INQUIRE TERMINAL command 485

TEXTPRINTST option
 INQUIRE TERMINAL command 485

Threadsafe
 EXEC CICS SPI commands 863

TIME option
 INQUIRE DISPATCHER
 command 298
 INQUIRE MONITOR command 384
 INQUIRE REQID command 425
 INQUIRE SYSTEM command 451
 SET DISPATCHER command 645
 SET SYSTEM command 720

timeout delete mechanism 294

TIMEOUT value
 INQUIRE TCPIPSERVICE
 command 474

TIMEOUTINT
 CEMT INQUIRE WEB 565
 SET WEB 769

TITLE option
 PERFORM DUMP command 587

TITLELENGTH option
 PERFORM DUMP command 587

TOTAL option
 INQUIRE JVMPOOL command 368

TOTALJVM option
 INQUIRE CLASSCACHE
 command 258

TPNAME option
 INQUIRE PARTNER command 399

TPNAMELEN option
 INQUIRE PARTNER command 399

TRACEDEST, INQUIRE command 505

TRACEDEST, SET command 742

TRACEFLAG, INQUIRE command 507

TRACEFLAG, SET command 746

TRACETYPE, INQUIRE command 509

TRACETYPE, SET command 748

TRACING option
 INQUIRE TASK command 462
 INQUIRE TERMINAL command 485
 INQUIRE TRANSACTION
 command 525
 SET TERMINAL command 740
 SET TRANSACTION command 758

trademarks 866

TRANCLASS option
 COLLECT STATISTICS command 33
 CREATE TRANCLASS
 command 110
 DISCARD TRANCLASS
 command 195
 INQUIRE TASK command 462
 INQUIRE TRANCLASS
 command 514
 INQUIRE TRANSACTION
 command 526
 PERFORM STATISTICS
 command 603
 SET TRANCLASS command 752
 SET TRANSACTION command 758

TRANCLASS, CREATE command 109

TRANCLASS, DISCARD command 194

TRANCLASS, INQUIRE command 512

TRANCLASS, SET command 751

TRANDUMP
 PERFORM STATISTICS
 command 603
 TRANDUMPCODE option
 COLLECT STATISTICS command 33
 INQUIRE TRANDUMPCODE
 command 516
 SET TRANDUMPCODE
 command 755
 TRANDUMPCODE, INQUIRE
 command 514
 TRANDUMPCODE, SET command 752
 TRANDUMPING option
 INQUIRE TRANDUMPCODE
 command 516
 SET TRANDUMPCODE
 command 755
 TRANSOLATE option
 INQUIRE SYSTEM command 451
 TRANPRIORITY option
 INQUIRE TASK command 462
 TRANSACTION option
 COLLECT STATISTICS command 33
 CREATE TRANSACTION
 command 113
 DISCARD TRANSACTION
 command 196
 INQUIRE ASSOCIATION
 command 212
 INQUIRE BRFACTILITY
 command 235
 INQUIRE TASK command 462
 INQUIRE TERMINAL command 485
 INQUIRE TRANSACTION
 command 526
 INQUIRE URIMAP command 560
 PERFORM STATISTICS
 command 603
 SET TRANSACTION command 759
 transaction security checking 16
 TRANSACTION, DISCARD
 command 195
 TRANSACTION, INQUIRE
 command 517
 TRANSACTION, SET command 756
 TRANSID option
 INQUIRE PROGRAM command 410
 INQUIRE REQID command 425
 INQUIRE REQUESTMODEL
 command 428
 INQUIRE TCPIP SERVICE
 command 474
 INQUIRE TSQNAME command 534
 INQUIRE TSQUEUE command 534
 INQUIRE UOW command 537
 INQUIRE UOWENQ command 548
 INQUIRE WORKREQUEST
 command 574
 translator 1
 TRANSMODE option
 INQUIRE EPADAPTER
 command 321
 TRIGGERLEVEL option
 INQUIRE TDQUEUE command 484
 SET TDQUEUE command 731
 TRPROF option
 INQUIRE TASK command 462
 TRPROF option (*continued*)
 INQUIRE TRANSACTION
 command 526
 TSMAININUSE option
 INQUIRE TEMPSTORAGE
 command 485
 TSMAINLIMIT option
 INQUIRE TEMPSTORAGE
 command 485
 TSMODEL option
 CREATE TSMODEL command 115
 DISCARD TSMODEL command 197
 TSMODEL, DISCARD command 196
 TSMODEL, INQUIRE command 527
 TSPOOL option
 INQUIRE TSPOOL command 530
 TSPOOL, INQUIRE command 530
 TSQ option
 INQUIRE DOCTEMPLATE
 command 305
 TSQNAME
 INQUIRE TSQNAME command 534
 TSQNAME, INQUIRE command 531
 TSQPREFIX option
 INQUIRE TCPIP SERVICE
 command 474
 TSQUEUE
 INQUIRE TSQUEUE command 534
 TSQUEUE option
 COLLECT STATISTICS command 33
 INQUIRE DOCTEMPLATE
 command 305
 PERFORM STATISTICS
 command 603
 TSQUEUE, INQUIRE command 531
 TSQUEUELIMIT option
 INQUIRE MONITOR command 384
 SET MONITOR command 697
 TSYSTEM option
 INQUIRE WORKREQUEST
 command 574
 TSTYPE option
 INQUIRE WORKREQUEST
 command 574
 TTSTATUS option
 INQUIRE TERMINAL command 485
 SET TERMINAL command 740
 TWAIT option of DSNCRCT macro
 TYPE=ENTRY macro 624
 TWASIZE option
 INQUIRE TASK command 462
 INQUIRE TRANSACTION
 command 526
 TX
 option of DSNCRCT macro 276
 TX option of DSNCRCT macro 624
 TXID option of DSNCRCT macro 276,
 288, 637
 TYPE option
 INQUIRE DOCTEMPLATE
 command 305
 INQUIRE FILE command 345
 INQUIRE REQUESTMODEL
 command 428
 INQUIRE TDQUEUE command 484
 INQUIRE UOWENQ command 548
 INQUIRE UOWLINK command 553
 TYPETERM option
 CREATE TYPETERM command 119
 TYPETERM, CREATE command 116

U

UCTRANST option
 INQUIRE TERMINAL command 485
 SET TERMINAL command 740
 UDSASIZE option
 INQUIRE SYSTEM command 451
 UNAUTH option
 INQUIRE CORBASERVER
 command 275
 UNAVAILABLE value 470
 UNLOCK command
 CSD 162
 UNREGISTERED value 471
 UOW option
 INQUIRE TASK command 462
 INQUIRE UOW command 537
 INQUIRE UOWENQ command 549
 INQUIRE UOWLINK command 553
 UOW, INQUIRE command 535
 UOWACTION option
 SET CONNECTION command 619
 SET IPCONN command 679
 UOWDSNFAIL, INQUIRE
 command 539
 UOWENQ, INQUIRE command 544
 UOWLINK option
 INQUIRE UOWLINK command 553
 SET UOWLINK command 762
 UOWLINK, INQUIRE command 550
 UOWSTATE option
 INQUIRE UOW command 537
 UPDATE option
 INQUIRE FILE command 346
 SET FILE command 662
 UPDATEMODEL option
 INQUIRE FILE command 346
 SET FILE command 662
 URID option
 INQUIRE EXCI command 326
 INQUIRE UOWLINK command 553
 URIMAP CVDA value
 EXTRACT STATISTICS
 command 210
 URIMAP option
 CREATE URIMAP command 122
 DISCARD URIMAP command 197
 INQUIRE ATOMSERVICE
 command 228
 INQUIRE URIMAP command 560
 INQUIRE WEBSERVICE
 command 570
 PERFORM STATISTICS
 command 603
 URIMAP, CREATE command 120
 URM option
 INQUIRE TCPIP SERVICE
 command 474
 SET TCPIP SERVICE command 727
 USAGE option
 INQUIRE URIMAP command 560

USECOUNT option
 INQUIRE EXITPROGRAM
 command 332
 INQUIRE PROGRAM command 410
USER
 option of DSNCRCT macro 288
USER value
 INQUIRE TCPIP SERVICE
 command 474
USERAREA option
 INQUIRE TERMINAL command 485
USERAREALEN option
 INQUIRE TERMINAL command 485
USERAUTH option
 INQUIRE IPCONN command 348
USERCORRDATA option
 INQUIRE ASSOCIATION
 command 212
 INQUIRE ASSOCIATION LIST
 command 222
USERDATA option
 ACQUIRE TERMINAL command 32
USERDATALEN option
 ACQUIRE TERMINAL command 33
USERDEFINE command
 CSD 163
USERID
 option of DSNCRCT macro 276
USERID option
 INQUIRE ASSOCIATION
 command 212
 INQUIRE BR FACILITY
 command 235
 INQUIRE REQID command 425
 INQUIRE TASK command 462
 INQUIRE TERMINAL command 485
 INQUIRE URIMAP command 560
USERID option of DSNCRCT
 macro 288, 637
USERNAME option
 INQUIRE TERMINAL command 485
USERSTATUS option
 INQUIRE TRACEFLAG
 command 508
 SET TRACEFLAG command 747

V

VALIDATIONST option
 INQUIRE TERMINAL command 485
 INQUIRE WEBSERVICE
 command 570, 769
 XMLTRANSFORM command 579,
 772
VALIDITY option
 INQUIRE DSNNAME command 311
VARIABLENAME option
 INQUIRE CAPDATAPRED
 command 243
 INQUIRE CAPINFOSRCE
 command 245
VERSION option
 INQUIRE EVENTBINDING
 command 324
 SET PROGRAM command 707
VFORMST option
 INQUIRE TERMINAL command 485

VOLUME, INQUIRE command 561
VOLUME, SET command 765
VTAM, INQUIRE command 561
VTAM, SET command 765

W

WAIT value
 INQUIRE TCPIP SERVICE
 command 474
WAITCAUSE option
 INQUIRE UOW command 538
WAITSTATE option
 INQUIRE UOW command 539
Web support
 INQUIRE transaction 564
 SET command 768
 WEB 768
WEBSERVICE CVDA value
 EXTRACT STATISTICS
 command 210
WEBSERVICE option
 CREATE WEBSERVICE
 command 124
 INQUIRE URIMAP command 560
 INQUIRE WEBSERVICE
 command 570, 769
 PERFORM STATISTICS
 command 603
WEBSERVICE, CREATE command 123
WEBSERVICE, DISCARD command 198
where-clause, CICS command format 6
WORKREQUEST option
 INQUIRE WORKREQUEST
 command 574
WORKREQUEST, INQUIRE
 command 571
WORKTYPE option
 INQUIRE WORKREQUEST
 command 574
 SET WORKREQUEST command 771
WSBIND option
 INQUIRE WEBSERVICE
 command 570
WSDLFILE option
 INQUIRE WEBSERVICE
 command 570

X

XCFGROUP option
 INQUIRE IRC command 359
XLNSTATUS option
 INQUIRE CONNECTION
 command 267
XLT option
 PERFORM SHUTDOWN
 command 597
XMLTRANSFORM command
 conditions 579, 773
XMLTRANSFORM CVDA value
 EXTRACT STATISTICS
 command 210
XMLTRANSFORM option
 INQUIRE ATOMSERVICE
 command 228

XMLTRANSFORM option (*continued*)
 PERFORM STATISTICS
 command 603
 XMLTRANSFORM command 773
XOPDIRECTST option
 INQUIRE WEBSERVICE
 command 570
XOPSUPPORTST option
 INQUIRE WEBSERVICE
 command 571
XRFSTATUS option
 INQUIRE SYSTEM command 452

Z

z/OS Communications Server 765
z/OS Communications Server option
 COLLECT STATISTICS command 33
 PERFORM STATISTICS
 command 603
ZCPTRACING option
 INQUIRE CONNECTION
 command 267
 INQUIRE TERMINAL command 485
 SET CONNECTION command 620
 SET TERMINAL command 741

Readers' Comments — We'd Like to Hear from You

CICS Transaction Server for z/OS
Version 4 Release 2
System Programming Reference

Publication No. SC34-7186-01

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your 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. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.
- Send a fax to the following number: +44 1962 816151
- Send your comments via email to: idrctf@uk.ibm.com

If you would like a response from IBM, please fill in the following information:

Name

Address

Company or Organization

Phone No.

Email address



Fold and Tape

Please do not staple

Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM United Kingdom Limited
User Technologies Department (MP095)
Hursley Park
Winchester
Hampshire
United Kingdom
SO21 2JN

Fold and Tape

Please do not staple

Fold and Tape



SC34-7186-01

