## Contents

Preface ................................................................. ix  
What this book is about .............................................. ix  
Who should read this book .......................................... ix  
What you need to know to understand this book ............... ix  
How to use this book ................................................ ix  
Notes on terminology .............................................. ix  

Summary of changes .................................................. xi  
Changes for CICS Transaction Server for z/OS, Version 3 Release 1 ......................................................... xi  
Changes for CICS Transaction Server for z/OS, Version 2 Release 3 .............................................................. xi  
Changes for CICS Transaction Server for z/OS, Version 2 Release 2 .............................................................. xii  
Changes for CICS Transaction Server for z/OS, Version 2 Release 1 .............................................................. xii  
Changes for the CICS Transaction Server for OS/390 Version 1 Release 3 edition ......................................................... xiii  
Changes for the CICS Transaction Server for OS/390 Version 1 Release 2 edition ......................................................... xiv  
Changes for the CICS Transaction Server for OS/390 Version 1 Release 1 edition ......................................................... xiv  

Chapter 1. Introduction to System Programming commands ........................................................... 1  
How this book is organized ............................................. 2  
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 ............................................................... 11  
COBOL argument values ............................................. 11  
C, and C++ argument values ........................................ 12  
PL/I argument values ................................................ 12  
Assembler-language argument values .............................. 13  
Argument lengths ..................................................... 14  
Null values ............................................................ 14  
Exception conditions .................................................. 15  
RESP and RESP2 options ............................................. 15  
Security checking ..................................................... 16  
The QUERY SECURITY command ................................... 18  
Inquiry commands ..................................................... 19  
Browsing resource definitions ...................................... 21  
Starting a browse .................................................... 21  
Retrieving the next resource ........................................ 22  
Ending the browse .................................................... 22  
Browse example ....................................................... 22  
Rules for browsing ................................................... 23  
Exception conditions for browsing ................................. 23  
SET commands ......................................................... 23  
Creating resource definitions ...................................... 24  
ATTRIBUTES option ................................................. 26  
Discarding resource definitions .................................... 27  
Exit-related commands .............................................. 28
INQUIRE SYSUDUMPCODE .................................. 314
INQUIRE SYSTEM ...................................... 317
INQUIRE TASK ......................................... 328
INQUIRE TASK LIST ................................... 339
INQUIRE TCLASS ....................................... 341
INQUIRE TCPIP ......................................... 342
INQUIRE TCPIPSERVICE ................................ 344
INQUIRE TDQUEUE ...................................... 350
INQUIRE TERMINAL .................................... 358
INQUIRE TRACEDEST ................................... 378
INQUIRE TRACEFLAG .................................. 381
INQUIRE TRACETYPE .................................. 383
INQUIRE TRANCLASS .................................. 386
INQUIRE TRANDUMPCODE ................................ 388
INQUIRE TRANSACTION ................................ 391
INQUIRE TSMODEL ..................................... 399
INQUIRE TSPOOL ....................................... 401
INQUIRE TSQUEUE / TSQNAME .......................... 403
INQUIRE UOW .......................................... 407
INQUIRE UOWDSNFAIL ................................ 412
INQUIRE UOWENQ ..................................... 417
INQUIRE UOWLINK ..................................... 423
INQUIRE URIMAP ....................................... 427
INQUIRE VOLUME ....................................... 432
INQUIRE VTAM .......................................... 433
INQUIRE WEB .......................................... 436
INQUIRE WEBSERVICE ................................ 437
INQUIRE WORKREQUEST ................................. 440
PERFORM CLASSCACHE .................................. 443
PERFORM CORBASERVER ................................ 447
PERFORM DELETESHIPPED ................................ 450
PERFORM DJAR .......................................... 451
PERFORM DUMP ......................................... 453
PERFORM ENDAFFINITY ................................ 456
PERFORM PIPELINE .................................... 458
PERFORM RESETTIME .................................... 460
PERFORM SECURITY REBUILD .......................... 461
PERFORM SHUTDOWN .................................... 463
PERFORM STATISTICS RECORD .......................... 466
RESYNC ENTRYNAME .................................... 471
SET AUTOINSTALL ....................................... 474
SET BRFACILITY ........................................ 476
SET CLASSCACHE ........................................ 478
SET CONNECTION ........................................ 480
SET CORBASERVER ..................................... 489
SET DB2CONN .......................................... 492
SET DB2ENTRY .......................................... 502
SET DB2TRAN .......................................... 508
SET DELETESHIPPED .................................... 510
SET DISPATCHER ........................................ 513
SET DSNNAME .......................................... 516
SET DUMPS ............................................. 525
SET ENQMODEL ......................................... 527
SET FILE ............................................... 529
SET HOST ............................................... 542
SET IRC .................................................. 544
Preface

What this book is about

This book describes the CICS® Transaction Server for z/OS® 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. Guidance information is in the CICS Application Programming Guide. For information about debugging CICS applications, see the CICS Problem Determination Guide.

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 manual 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.

Notes on terminology

- ‘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 ENTRYNNAME commands and some of the front-end programming interface (FEPI) commands.
- MB equals 1 048 576 bytes.
- KB equals 1024 bytes.
Summary of changes

This book is based on the CICS System Programming Reference for CICS
Transaction Server for z/OS, Version 2 Release 1, SC34-5726-00. Changes from
that edition are marked by vertical bars in the left margin.

This part lists briefly the changes that have been made for the following recent
releases:

Changes for CICS Transaction Server for z/OS, Version 3 Release 1

The more significant changes for this edition are:

• New commands:
  – “CREATE PIPELINE” on page 83
  – “CREATE URIMAP” on page 118
  – “CREATE WEBSERVICE” on page 121
  – “DISCARD PIPELINE” on page 142
  – “DISCARD WEBSERVICE” on page 156
  – “EXTRACT STATISTICS” on page 166
  – “INQUIRE PIPELINE” on page 282
  – “PERFORM PIPELINE” on page 458
  – “SET PIPELINE” on page 558

Because of the removal of support for HPJ (Hot-pooling) for Java program objects,
the following options are removed:

• The HOTPOOL option is removed from the EXEC CICS CREATE PROGRAM
  command.
• The HOTPOOLID option is removed from the EXEC CICS INQUIRE
  PROGRAM command.
• The HOTPOOLID option is removed from the EXEC CICS SET PROGRAM
  command.
• The ACTHPTCBS and MAXHPTCBS options are removed from the EXEC CICS
  INQUIRE DISPATCHER command.
• The MAXHPTCBS option is removed from the EXEC CICS SET DISPATCHER
  command.
• The CVDFs HOTPOOL (1065) and NOTHOTPOOL (1066) are deleted.

Changes for CICS Transaction Server for z/OS, Version 2 Release 3

The more significant changes for this edition are:

• New commands:
  – “INQUIRE WORKREQUEST” on page 440 added.
  – “SET WORKREQUEST” on page 635 added.
  – “INQUIRE JVM” on page 262 added.
  – “INQUIRE JVMPROFILE” on page 267 added.
  – “INQUIRE CLASSCACHE” on page 180 added.
  – “SET CLASSCACHE” on page 478 added.
  – “PERFORM CLASSCACHE” on page 443 added.

• New options:
  – “DEBUGTOOL” on page 320 added to INQUIRE SYSTEM
  – “DEBUGTOOL” on page 575 added to SET SYSTEM
  – “ASSERTED” on page 194 added to INQUIRE CORBASERVER
  – “PRIVACY” on page 347 added to INQUIRE TCPIPSERVICE
Changes for CICS Transaction Server for z/OS, Version 2 Release 2

The more significant changes for this edition are:

- New commands:
  - “INQUIRE DISPATCHER” on page 214 is a new command
  - “SET DISPATCHER” on page 513 is a new command

- New options:
  - “DB2GROUPID” on page 201 added to INQUIRE DB2CONN
  - “RESYNCMEMBER” on page 202 added to INQUIRE DB2CONN
  - “PLAN” on page 210 added to INQUIRE DB2TRAN
  - “PLANEXITNAME” on page 210 added to INQUIRE DB2TRAN
  - “SCAN” on page 448 added to PERFORM CORBASERVER
  - “DB2GROUPID” on page 496 added to SET DB2CONN
  - “RESYNCMEMBER” on page 498 added to SET DB2CONN

- Technical changes:
  - In “INQUIRE DB2CONN” on page 197, the definitions of parameters DB2ID, PRIORITY, TCBLIMIT and TCBS have changed.
  - In “INQUIRE DB2ENTRY” on page 205, the definition of the PRIORITY parameter has changed.
  - In “SET DB2CONN” on page 492, the definitions of parameters DB2ID, PRIORITY, TCBLIMIT and TCBS have changed.
  - In “SET DB2ENTRY” on page 502, the definition of the PRIORITY parameter has changed.

Changes for CICS Transaction Server for z/OS, Version 2 Release 1

The more significant changes for this edition are:

- New commands:
  - “SET DB2CONN” on page 492
  - “CREATE DJAR” on page 63
  - “DISCARD CORBASERVER” on page 130
  - “DISCARD DJAR” on page 134
  - “INQUIRE BEAN” on page 173
  - “INQUIRE CORBASERVER” on page 194
  - “INQUIRE DJAR” on page 217
  - “INQUIRE JVMPOOL” on page 265
  - “PERFORM CORBASERVER” on page 447
  - “PERFORM DJAR” on page 451
  - “SET CORBASERVER” on page 489
  - “SET JVMPOOL” on page 549

- New options:
  - “CORBASERVER” on page 42 added to COLLECT STATISTICS
  - “JVMPOOL” on page 43 added to COLLECT STATISTICS
  - “REQUESTMODEL” on page 44 added to COLLECT STATISTICS
  - “TCPIP” on page 45 added to COLLECT STATISTICS
  - “HOTPOOLSING” added to INQUIRE PROGRAM. (This option was removed for CICS Transaction Server for z/OS Version 3 Release 1.)
  - “JVMPROFILE” on page 291 added to INQUIRE PROGRAM
Changes for the CICS Transaction Server for OS/390 Version 1 Release 3 edition

• The system programming interface commands:
  – EXEC CICS CREATE
  – EXEC CICS DISCARD
  – EXEC CICS INQUIRE
  – EXEC CICS SET

are enhanced to provide function for the management of the following new resources:
  – DOCTEMPLATE
  – ENQMODEL
  – PROCESSTYPE
  – REQUESTMODEL
  – TCPIPSERVICE
  – TSMODEL

• New CVDA values for this release:
  – “NOTAPPLIC” on page 293 added to RUNTIME in INQUIRE PROGRAM
  – “OTSCOORD” on page 410 added to WAITCAUSE in INQUIRE UOW

• New conditions for this release:
  – “NORMAL” on page 584 added to SET TCPIP
  – “NOTSUPERUSER” on page 584 added to SET TCPIP
The EXEC CICS INQUIRE TDQUEUE command is extended to return the name of the MEMBER when the DDNAME references a partitioned data set.

The system programming interface commands:
- EXEC CICS CREATE
- EXEC CICS DISCARD
- EXEC CICS INQUIRE
- EXEC CICS SET

are enhanced to provide function for the management of programs running under a JVM.

Changes for the CICS Transaction Server for OS/390 Version 1 Release 2 edition

The major changes to CICS that affect CICS Transaction Server for OS/390®, Version 1 Release 2 are:

- Addition of BRIDGE and IDENTIFIER to the INQUIRE TASK command to support the 3270 bridge.
- Addition of BREXIT and FACILITYLIKE to the INQUIRE TRANSACTION command to support the 3270 bridge.
- The CICS DB2 attachment facility is enhanced to provide resource definition online (RDO) support for DB2 resources as an alternative to resource control table (RCT) definitions. The system programming interface is enhanced to provide function for the management of the CICS DB2 attachment facility. The following new commands are added for DB2 resource definitions, DB2CONN, DB2ENTRY, and DB2TRAN:
  - EXEC CICS CREATE
  - EXEC CICS DISCARD
  - EXEC CICS INQUIRE
  - EXEC CICS SET

The following commands are extended:
- EXEC CICS INQUIRE TASK returns the name of the DB2 plan used for the task.
- EXEC CICS INQUIRE SYSTEM returns the name of the DB2CONN currently installed.

Changes for the CICS Transaction Server for OS/390 Version 1 Release 1 edition

The major changes to CICS Transaction Server for OS/390 Release 1 that affect this book are:

- New DISCARD CONNECTION, DISCARD JOURNALMODEL, DISCARD JOURNALNAME, DISCARD TDQUEUE, DISCARD TERMINAL, INQUIRE JOURNALMODEL, INQUIRE JOURNALNAME (replacing INQUIRE JOURNALNUM), INQUIRE STREAMNAME, INQUIRE UOW, INQUIRE UOWDSNFAIL, INQUIRE UOWENQ, INQUIRE UOWLINK, PERFORM ENDAFFINITY, SET JOURNALNAME (replacing SET JOURNALNUM), SET UOW, and SET UOWLINK commands
- New or changed options for COLLECT STATISTICS, DISABLE PROGRAM, ENABLE PROGRAM, INQUIRE CONNECTION, INQUIRE DSNAME, INQUIRE EXITPROGRAM, INQUIRE FILE, INQUIRE NETNAME, INQUIRE SYSTEM, INQUIRE TASK, INQUIRE TDQUEUE, INQUIRE TERMINAL, INQUIRE
TRANSACTION, INQUIRE TSQUEUE, PERFORM SHUTDOWN, PERFORM STATISTICS RECORD, RESYNC ENTRYYNAME, SET CONNECTION, SET DSNAMEx, and SET FILE commands

• Addition of the CREATE commands
• INQUIRE VOLUME and SET VOLUME commands now obsolete
• Removal of local DL/I options
• New conditions on the SET TDQUEUE command

Other changes made to this book include:
Chapter 1. Introduction to System Programming commands

This book describes the CICS system programming interface (SPI) commands. These 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 a companion manual to this one, the CICS Application Programming Reference manual. A third manual, the CICS Application Programming Guide 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 SYSSID 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

1. Users of much earlier releases of CICS may be familiar with the old programmable interface to the master terminal program (DFHEMTA). Its use is still supported, though the documentation is available only in the CICS Customization Guide for releases prior to CICS/ESA Version 3.

You cannot use DPL to link to the CICS master terminal program, DFHEMTA. The addresses passed as parameters to DFHEMTA are valid only in the region that issues the EXEC CICS LINK command, which means you cannot route a DFHEMTA request to a remote CICS system. The same restriction also applies to the programmable interface to the RDO transaction, CEDA, invoked through program DFHEDAP.
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, where the DFHTST TYPE=REMOTE entry is
provided, are an exception.

- Additional security checking is available for SPI commands, as explained on
  page “The format of SPI commands” on page 3.
- 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.
These notes begin on page 19.

---

### How this book is organized

The next topics in this chapter explains how SPI commands are written, including
information on:
- Command format
- Syntax diagrams
- Argument types and values
- Data formats
- Language-specific notes

There is general information that applies to all SPI commands:
- Exception conditions
- Response codes (RESP and RESP2)
- Security checking

And there is specific information for these groups of commands:
- Inquiries and browsing
- SET commands
- CREATE commands
- DISCARD commands
- Exit-related commands

You should read this material even if you already know how to write CICS
commands. It begins on page 15.

Chapter 2 of this book describes the SPI commands individually, in alphabetical
order. Descriptions begin with a brief statement of what the command does,
followed by a syntax diagram and general information about usage. Command
options are described next, also in alphabetical order. A list of the exception
conditions that can arise during execution of the command follows and, in some
cases, there are also examples of usage.

---

### Command format

SPI commands are written in the same way as API commands. They begin with the
words EXECUTE CICS (usually abbreviated EXEC CICS), followed by the
command name, a verb or verb-and-option combination such as:

- 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/370™™ 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:

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

becomes

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

in COBOL, and

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

in C/370 or PL/I.

For more information about translating the commands, see the CICS Application Programming Guide for translator options, and the CICS Application Programming Guide for the job control language.

---

**CICS syntax notation**

Throughout this book, 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.

**Note:** 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 15 for additional SPI details.)

You read the diagram by following the arrows from left to right, using these conventions:
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="symbolA.png" alt="Symbol A" /></td>
<td>A set of alternatives—one of which you <strong>must</strong> code.</td>
</tr>
<tr>
<td><img src="symbolB.png" alt="Symbol B" /></td>
<td>A set of alternatives—one of which you <strong>must</strong> code. You <strong>may</strong> code more than one of them, in any sequence.</td>
</tr>
<tr>
<td><img src="symbolC.png" alt="Symbol C" /></td>
<td>A set of alternatives—one of which you <strong>may</strong> code.</td>
</tr>
<tr>
<td><img src="symbolD.png" alt="Symbol D" /></td>
<td>A set of alternatives — any number (including none) of which you may code once, in any sequence.</td>
</tr>
<tr>
<td><img src="symbolE.png" alt="Symbol E" /></td>
<td>Alternatives where A is the default.</td>
</tr>
<tr>
<td><img src="symbolF.png" alt="Symbol F" /></td>
<td>Use with the named section in place of its name. <strong>Name:</strong></td>
</tr>
<tr>
<td><img src="symbolG.png" alt="Symbol G" /></td>
<td>Code exactly as shown. <strong>Punctuation and uppercase characters</strong></td>
</tr>
<tr>
<td><img src="symbolH.png" alt="Symbol H" /></td>
<td>Code your own text, as appropriate (for example, name). <strong>Lowercase characters</strong></td>
</tr>
</tbody>
</table>

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 wish, 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 wish 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 wish 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 11.

- 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/ESA™ 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 *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 the FREEMAIN command in the *CICS Application Programming Reference* manual for more information about the distinction between data and pointers.

CICS-value data areas (CVDAs)

A *CVDA* (CICS-value data area) is still another special case: it 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 simply 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 ENQUIRE PROGRAM (TAXPGM) EXECUTIONSET (TAXAPI) END-EXEC.
IF TAXAPI = DFHVALUE(FULLAPI) PERFORM STND-INIT ELSE PERFORM REMOTE-INIT.

Appendix A, “CICS-value data areas used by all commands,” on page 637 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 which show the use of CVDAs and the DFHVALUE function. In each case, the code provided:

- 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

```cobol
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

```c
#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

```pli
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
NOTDELETEABLE
OPENLAB EXEC CICS RETURN
```

Data types

For most arguments, CICS uses one of five data types (formats):

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

The first four are all used for numeric data, but they differ in length and internal format; the last is for text. The names used in this book are those used in assembler language.

Data-areas and data-values may require any of these formats. The option text tells you which one to use. CVDAs 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 some language-specific information may be found in the CICS Application Programming Guide.

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.
### 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 table below shows how to define the correct data type.

<table>
<thead>
<tr>
<th>Data type</th>
<th>C definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Halfword binary</td>
<td>short int</td>
</tr>
<tr>
<td>Fullword binary (including CVDA)</td>
<td>long int</td>
</tr>
<tr>
<td>Doubleword binary</td>
<td>long long int</td>
</tr>
<tr>
<td>Character string (n characters long)</td>
<td>unsigned char[n]</td>
</tr>
<tr>
<td>Packed decimal</td>
<td>Not used—see note</td>
</tr>
</tbody>
</table>

**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 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 table below shows how to define the correct data type:

<table>
<thead>
<tr>
<th>Data type</th>
<th>PL/I definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Halfword binary</td>
<td>FIXED BIN(15)</td>
</tr>
<tr>
<td>Fullword binary (including CVDA)</td>
<td>FIXED BIN(31)</td>
</tr>
<tr>
<td>Doubleword binary</td>
<td>FIXED BIN(63)</td>
</tr>
<tr>
<td>Pointer</td>
<td>POINTER</td>
</tr>
</tbody>
</table>
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.

The translator generates ENTRY statements when it translates your CICS commands to PL/I CALLs 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 PL/I’s 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 length, and so this also is usually unintended.

**Assembler-language argument values**

In assembler 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 doesn’t 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. (See the “Inquiry commands” on page 19 and “SET commands” on page 23 for more about these uses.)

The null value for each data type is listed below:

<table>
<thead>
<tr>
<th>Data type</th>
<th>Null value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Character string (n characters long)</td>
<td>n blanks (X’40’)</td>
</tr>
<tr>
<td>Halfword binary</td>
<td>-1 (X’FFFF’)</td>
</tr>
<tr>
<td>Fullword binary</td>
<td>-1 (X’FFFFFFFF’)</td>
</tr>
<tr>
<td>Doubleword binary</td>
<td>-1 (X’FFFFFFFFFFFFFFFF’)</td>
</tr>
<tr>
<td>Pointer (address)</td>
<td>X’FF000000’</td>
</tr>
<tr>
<td>CVDA (in receiver option)</td>
<td>DFHVALUE(NOTAPPLIC) (-1 or X’FFFFFFFF’)</td>
</tr>
<tr>
<td>CVDA (in sender option)</td>
<td>DFHVALUE(IGNORE) (-1 or X’FFFFFFFF’)</td>
</tr>
</tbody>
</table>
Exception conditions

CICS does not distinguish between SPI and API commands in the flow of control after it processes a command. You should 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 671.

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 similar to 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 671 lists the correspondence, but you should 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:

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

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

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

and then test explicitly for a remote file:
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 need 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/370: long int
- PL/I: FIXED BIN(31)
- Assembler: F

### Security checking

CICS uses an external security manager, such as RACF®, to perform security checking. The facilities CICS provides are summarized in this section, but you will need to refer to other manuals for full information. The [CICS RACF Security Guide](#) is the most comprehensive reference; it describes in detail how CICS uses RACF facilities to implement security. System and resource definition options that govern security are described in the [CICS System Definition Guide](#) and the [CICS Resource Definition Guide](#), respectively, and the [CICS Customization Guide](#) contains general information on CICS security.

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 if, and 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. SPI commands are the ones described in this book, and are the same ones that require the SP option for translation (see page "The format of SPI commands" on page 3).

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

CICS returns a RESP2 value of 102 if the check fails. (Additional checks on the assigned user occur on SET TDQUEUE ATIUSERID, 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
- XTST for temporary storage queues
- XDB2 for DB2 entries and transactions

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 simply skips resources that would fail the resource check on an ordinary INQUIRE (see “Rules for browsing” on page 23 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.

<table>
<thead>
<tr>
<th>Resource (system initialization parameter)</th>
<th>Commands</th>
</tr>
</thead>
</table>
| 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)    
<p>|                                          | SET FILE (U)        |</p>
<table>
<thead>
<tr>
<th>Resource (system initialization parameter)</th>
<th>Commands</th>
</tr>
</thead>
</table>
| Journals (XJCT option)                    | COLLECT STATISTICS JOURNALNAME (R)  
|                                           | COLLECT STATISTICS JOURNALNUM (R)  
|                                           | DISCARD JOURNALNAME (A)            
|                                           | INQUIRE JOURNALNAME (R)            
|                                           | SET JOURNALNAME (U)                |
| Programs Map sets                          | COLLECT STATISTICS PROGRAM (R)     
| Partition sets (XPPT option)               | 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 TCLASS (A)                  
|                                           | CREATE TRANCLASS (A)               
|                                           | DISCARD TCLASS (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)                    |

**Authtype** checking applies to DB2CONNs, DB2ENTRYs, and DB2TRANs only. For more information, see the CICS 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 manual.

Inquiry commands

The system programming commands allow you to inquire about the definition and status of most of the resources defined to CICS, and about many elements of the CICS system as well. The resources about which you can inquire are:

- Autoinstall terminal models (AUTINSTMODEL)
- Bridge facilities used by the 3270 bridge (BRFACILITY)
- Enterprise beans (BEAN)
- Coupling facility data table server connections (CFDTPOOL)
- Connections (CONNECTION and UOWLINK)
- CorbaServers (CORBASESERVER)
- DB2 connections (DB2CONN)
- DB2 entries (DB2ENTRY)
- DB2 transactions (DB2TRAN)
- DOC templates (DOCTEMPLATE)
- Deployed JAR files (DJAR)
- Exits (EXITPROGRAM)
- External data sets (DSNAME and UOWDSNFLAIL)
- Files (FILE)
- Installed enterprise bean (BEAN)
- Journals (JOURNALNAME and JOURNALMODEL)
- JVMs (JVM)
- JVM profiles (JVMPROFILE)
- Log streams (STREAMNAME)
- Map sets (PROGRAM)
- Partition sets (PROGRAM)
- Partners (PARTNER)
- Pipelines (PIPELINE)
- Process type (PROCESSTYPE)
- Profiles (PROFILE)
- Programs (PROGRAM)
- Request models (REQUESTMODEL)
- Session groups (MODENAME)
- Storage subpools (SUBPOOL)
- System dump codes (SYSDUMPCODE)
- TCP/IP services (TCPIPSERVICE)
- Temporary storage queues (TSQUEUE and TSONAME)
- Temporary storage pools (TSPOOL)
- Temporary storage models (TSMODEL)
- Terminals (TERMINAL, NETNAME)
- Transaction classes (TCLASS, TRANCLASS)
- Transaction dump codes (TRANDUMPCODE)
- Transactions (TRANSACTION)
- Transient data queues (TDQUEUE)
- URIMAP resource definitions (URIMAP)
- Virtual hosts (HOST)
- Web services (WEBSERVICE)

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 in this manual notes the corresponding definition.
option. Consequently, if you need additional information about the meaning of an
option value, it is often helpful to refer to the definition of the resource in the CICS
Resource Definition Guide.

The system elements about which you can inquire are:
Autoinstall for terminals (AUTOINSTALL)
Dispatcher (DISPATCHER)
Dump data sets (DUMPDS)
Enqueues (UOWENQ)
Interregion communication (IRC)
Monitor (MONITOR)
MVS TCBs (MVSTCB)
JVM pool (JVMPOOL)
Requests (REQID)
Shared class cache (CLASSCACHE)
Shipped terminal status (DELETISHED)
Statistics (STATISTICS)
Storage (STORAGE)
System status (SYSTEM)
Tasks (TASK, TASK LIST)
TCP/IP (TCPIP)
Tracing (TRACEDEST, TRACEFLAG, TRACETYPE)
Transactional EXCI (RRMS)
Units of work (UOW, UOWDSNFAIL, UOWENQ, UOWLINK)
VTAM®
Web interface (WEB)
Work requests (WORKREQUEST)

Most of these elements correspond to system initialization parameters. If you need
more information about them, see the system initialization parameters discussion in
the CICS System Definition Guide.

Certain considerations apply to all of the inquiry commands, which are principally
the INQUIREs, 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 inquiring about a resource, system setting, or system
  component does not get exclusive control of the object of the inquiry. The
  information returned may be changed by another task or system event at any
time. The resource currently being inquired on should not be deleted since the
current resource is used to position to the next resource on a subsequent
GETNEXT command. Only after the subsequent GETNEXT command can the
resource be deleted, since it is no longer required for positioning within this
browse request.
• **Browsing**: Resources defined in the first (resource) list can be retrieved
  sequentially, as explained in “Browsing resource definitions” on page 21.
• **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
for extrapartition queues only.
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, and the individual command syntax discussions in the next section describe them in this form.

However, there is another form that enables you to browse through some or all of the definitions of a given type. The resource types that you are allowed to browse are those in the first list in "Inquiry commands" on page 19 plus requests (REQIDs) and units of work (UOWs, UOWDSNFAILs, UOWENQs, and UOWLINKs).

Starting a browse

A browse involves three steps. First, 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:

```
Browse START
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:

```
START browse AT
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:

```
Browse NEXT
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, thus:

```
Browse END
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(RECVRSTAT)
      FWDRECSTATUS(FWDSSTAT)
      RESP(RESP CODE) END-EXEC
   IF RESP CODE = DFHRESP(NORMAL)
IF OPENSTAT = DFHVALUE(OPEN)
    CALL RCVY-RTN USING RCVRSTAT FWDSRAT
END-IF
ELSE CALL ERROR-RTN END-IF
END-PERFORM.
EXEC CICS INQUIRE FILE END END-EXEC.

Rules for browsing
In addition to the syntax changes described above, there are some rules you should note 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.

   Note: 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:

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)
LSRPOOLID(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 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:
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 syncpoints 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 syncpointing. 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 (PLTP) as well as during normal CICS execution.

**ATTRIBUTES option**

The specifics of the resource definition that a CREATE 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 command is executed, and consequently must consist entirely of text, rather than variable names, in a single string. The syntax within the string is provided for each CREATE 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 the CICS 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
    - A character string of length n or, where the argument can be of variable length, of maximum length n.
  - hex
    - 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. In all cases, you should refer to the CICS Resource Definition Guide for specific rules about the argument values.

- 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:
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, because CREATEs do not use the CSD file.)

For some resource types, you can use defaults for all attributes. If you wish 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, “EXEC CICS CREATE RESP2 values,” on page 681 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 CEDA transaction, or by an automatic installation process.

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

- Autoinstall models for terminals
- Connections
- CorbaServer (CORBASERVER)
- DB2Conns
- DB2Entry
- DB2Trans
- Deployed JAR files (DJAR)
- Document templates
- ENQ models
- Files
- Journals and journal models
- Partners
- PIPELINEs
- Process types
- Profiles
- Programs, map sets, and partition sets
- Request models
- TCP/IP service
- Temporary storage queue models
- Terminals
- Transaction classes
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, CICS syncpoints your task" on page 25 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 thus are 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 may be used both by an exit and a user transaction, in fact). 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 697.

For similar statements about the application programming interface (API), see the CICS Application Programming Reference manual.
# Chapter 2. System commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACQUIRE TERMINAL</td>
<td>35</td>
</tr>
<tr>
<td>COLLECT STATISTICS</td>
<td>38</td>
</tr>
<tr>
<td>CREATE CONNECTION</td>
<td>48</td>
</tr>
<tr>
<td>CREATE CORBASERVER</td>
<td>52</td>
</tr>
<tr>
<td>CREATE DB2CONN</td>
<td>55</td>
</tr>
<tr>
<td>CREATE DB2ENTRY</td>
<td>58</td>
</tr>
<tr>
<td>CREATE DB2TRAN</td>
<td>61</td>
</tr>
<tr>
<td>CREATE DJAR</td>
<td>63</td>
</tr>
<tr>
<td>CREATE DOCTEMPLATE</td>
<td>65</td>
</tr>
<tr>
<td>CREATE ENQMODEL</td>
<td>67</td>
</tr>
<tr>
<td>CREATE FILE</td>
<td>69</td>
</tr>
<tr>
<td>CREATE JOURNALMODEL</td>
<td>72</td>
</tr>
<tr>
<td>CREATE LSRPOOL</td>
<td>74</td>
</tr>
<tr>
<td>CREATE MAPSET</td>
<td>77</td>
</tr>
<tr>
<td>CREATE PARTITIONSET</td>
<td>79</td>
</tr>
<tr>
<td>CREATE PARTNER</td>
<td>81</td>
</tr>
<tr>
<td>CREATE PIPELINE</td>
<td>83</td>
</tr>
<tr>
<td>CREATE PROCESSTYPE</td>
<td>85</td>
</tr>
<tr>
<td>CREATE PROFILE</td>
<td>87</td>
</tr>
<tr>
<td>CREATE PROGRAM</td>
<td>89</td>
</tr>
<tr>
<td>CREATE REQUESTMODEL</td>
<td>92</td>
</tr>
<tr>
<td>CREATE SESSIONS</td>
<td>94</td>
</tr>
<tr>
<td>CREATE TCPIPSERVICE</td>
<td>97</td>
</tr>
<tr>
<td>CREATE TERMINAL</td>
<td>100</td>
</tr>
<tr>
<td>CREATE TCQUEUE</td>
<td>104</td>
</tr>
<tr>
<td>CREATE TRANCLASS</td>
<td>107</td>
</tr>
<tr>
<td>CREATE TRANSACTION</td>
<td>109</td>
</tr>
<tr>
<td>CREATE TSMODEL</td>
<td>112</td>
</tr>
<tr>
<td>CREATE TYPETERM</td>
<td>114</td>
</tr>
<tr>
<td>CREATE URIMAP</td>
<td>118</td>
</tr>
<tr>
<td>CREATE WEBSERVICE</td>
<td>121</td>
</tr>
<tr>
<td>DISABLE PROGRAM</td>
<td>123</td>
</tr>
<tr>
<td>DISCARD AUTINSTMODEL</td>
<td>127</td>
</tr>
<tr>
<td>DISCARD CONNECTION</td>
<td>128</td>
</tr>
<tr>
<td>DISCARD CORBASERVER</td>
<td>130</td>
</tr>
<tr>
<td>DISCARD DB2CONN</td>
<td>131</td>
</tr>
<tr>
<td>DISCARD DB2ENTRY</td>
<td>132</td>
</tr>
<tr>
<td>DISCARD DB2TRAN</td>
<td>133</td>
</tr>
<tr>
<td>DISCARD DJAR</td>
<td>134</td>
</tr>
<tr>
<td>DISCARD DOCTEMPLATE</td>
<td>135</td>
</tr>
<tr>
<td>DISCARD ENQMODEL</td>
<td>136</td>
</tr>
<tr>
<td>DISCARD FILE</td>
<td>137</td>
</tr>
<tr>
<td>DISCARD JOURNALMODEL</td>
<td>138</td>
</tr>
<tr>
<td>DISCARD JOURNALNAME</td>
<td>139</td>
</tr>
<tr>
<td>DISCARD PARTNER</td>
<td>141</td>
</tr>
<tr>
<td>DISCARD PIPELINE</td>
<td>142</td>
</tr>
<tr>
<td>DISCARD PROCESSTYPE</td>
<td>143</td>
</tr>
<tr>
<td>DISCARD PROFILE</td>
<td>144</td>
</tr>
<tr>
<td>DISCARD PROGRAM</td>
<td>145</td>
</tr>
<tr>
<td>DISCARD REQUESTMODEL</td>
<td>146</td>
</tr>
<tr>
<td>DISCARD TCPIPSERVICE</td>
<td>147</td>
</tr>
<tr>
<td>DISCARD TCQUEUE</td>
<td>148</td>
</tr>
<tr>
<td>DISCARD TERMINAL</td>
<td>150</td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 1977, 2011
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISCARD TRANCLASS</td>
<td>152</td>
</tr>
<tr>
<td>DISCARD TRANSACTION</td>
<td>153</td>
</tr>
<tr>
<td>DISCARD TMODEL</td>
<td>154</td>
</tr>
<tr>
<td>DISCARD URIMAP</td>
<td>155</td>
</tr>
<tr>
<td>DISCARD WEBSERVICE</td>
<td>156</td>
</tr>
<tr>
<td>ENABLE PROGRAM</td>
<td>157</td>
</tr>
<tr>
<td>EXTRACT EXIT</td>
<td>164</td>
</tr>
<tr>
<td>EXTRACT STATISTICS</td>
<td>166</td>
</tr>
<tr>
<td>INQUIRE AUTINSTMODEL</td>
<td>170</td>
</tr>
<tr>
<td>INQUIRE AUTOINSTALL</td>
<td>171</td>
</tr>
<tr>
<td>INQUIRE BEAN</td>
<td>173</td>
</tr>
<tr>
<td>INQUIRE BRFACILITY</td>
<td>175</td>
</tr>
<tr>
<td>INQUIRE CFDTPOOL</td>
<td>178</td>
</tr>
<tr>
<td>INQUIRE CLASSCACHE</td>
<td>180</td>
</tr>
<tr>
<td>INQUIRE CONNECTION</td>
<td>183</td>
</tr>
<tr>
<td>INQUIRE CORBASERVER</td>
<td>194</td>
</tr>
<tr>
<td>INQUIRE DB2CONN</td>
<td>197</td>
</tr>
<tr>
<td>INQUIRE DB2ENTRY</td>
<td>205</td>
</tr>
<tr>
<td>INQUIRE DB2TRAN</td>
<td>210</td>
</tr>
<tr>
<td>INQUIRE DELETISHED</td>
<td>212</td>
</tr>
<tr>
<td>INQUIRE DISPATCH</td>
<td>214</td>
</tr>
<tr>
<td>INQUIRE DJAR</td>
<td>217</td>
</tr>
<tr>
<td>INQUIRE DOCTEMPLATE</td>
<td>219</td>
</tr>
<tr>
<td>INQUIRE DSNAME</td>
<td>222</td>
</tr>
<tr>
<td>INQUIRE DUMPDS</td>
<td>229</td>
</tr>
<tr>
<td>INQUIRE ENQ</td>
<td>231</td>
</tr>
<tr>
<td>INQUIRE ENQMODEL</td>
<td>232</td>
</tr>
<tr>
<td>INQUIRE EXCI</td>
<td>234</td>
</tr>
<tr>
<td>INQUIRE EXITPROGRAM</td>
<td>236</td>
</tr>
<tr>
<td>INQUIRE FILE</td>
<td>242</td>
</tr>
<tr>
<td>INQUIRE HOST</td>
<td>254</td>
</tr>
<tr>
<td>INQUIRE IRC</td>
<td>256</td>
</tr>
<tr>
<td>INQUIRE JOURNALMODEL</td>
<td>257</td>
</tr>
<tr>
<td>INQUIRE JOURNALNAME</td>
<td>259</td>
</tr>
<tr>
<td>INQUIRE JOURNALNUM</td>
<td>261</td>
</tr>
<tr>
<td>INQUIRE JVM</td>
<td>262</td>
</tr>
<tr>
<td>INQUIRE JVMPOOL</td>
<td>265</td>
</tr>
<tr>
<td>INQUIRE JVMPROFILE</td>
<td>267</td>
</tr>
<tr>
<td>INQUIRE MODENAME</td>
<td>269</td>
</tr>
<tr>
<td>INQUIRE MONITOR</td>
<td>272</td>
</tr>
<tr>
<td>INQUIRE MVSTCB</td>
<td>276</td>
</tr>
<tr>
<td>INQUIRE NETNAME</td>
<td>278</td>
</tr>
<tr>
<td>INQUIRE PARTNER</td>
<td>280</td>
</tr>
<tr>
<td>INQUIRE PIPELINE</td>
<td>282</td>
</tr>
<tr>
<td>INQUIRE PROCESSSTYPE</td>
<td>284</td>
</tr>
<tr>
<td>INQUIRE PROFILE</td>
<td>286</td>
</tr>
<tr>
<td>INQUIRE PROGRAM</td>
<td>287</td>
</tr>
<tr>
<td>INQUIRE REQID</td>
<td>296</td>
</tr>
<tr>
<td>INQUIRE REQUESTMODEL</td>
<td>301</td>
</tr>
<tr>
<td>INQUIRE RRMS</td>
<td>303</td>
</tr>
<tr>
<td>INQUIRE STATISTICS</td>
<td>304</td>
</tr>
<tr>
<td>INQUIRE STORAGE</td>
<td>307</td>
</tr>
<tr>
<td>INQUIRE STREAMNAME</td>
<td>310</td>
</tr>
<tr>
<td>INQUIRE SUBPOOL</td>
<td>312</td>
</tr>
<tr>
<td>INQUIRE SYSDUMPCODE</td>
<td>314</td>
</tr>
<tr>
<td>INQUIRE SYSTEM</td>
<td>317</td>
</tr>
</tbody>
</table>
INQUIRE TASK ........................................ 328
INQUIRE TASK LIST .................................. 339
INQUIRE TCLASS .................................... 341
INQUIRE TCPIP ...................................... 342
INQUIRE TCPIPSERVICE .......................... 344
INQUIRE TDQUEUE ................................ 350
INQUIRE TERMINAL ................................ 358
INQUIRE TRACEDEST ............................. 378
INQUIRE TRACEFLAG ............................. 381
INQUIRE TRACETYPE ............................. 383
INQUIRE TRANCLASS ............................. 386
INQUIRE TRANDUMPCODE ....................... 388
INQUIRE TRANSACTION .......................... 391
INQUIRE TSMODEL ................................ 399
INQUIRE TSPOOL .................................. 401
INQUIRE TSQUEUE / TSQNAME ................. 403
INQUIRE UOW ..................................... 407
INQUIRE UOWDSNFAIL .......................... 412
INQUIRE UOWENQ ................................ 417
INQUIRE UOWLINK ............................... 423
INQUIRE URIMAP .................................. 427
INQUIRE VOLUME .................................. 432
INQUIRE VTAM ..................................... 433
INQUIRE WEB ...................................... 436
INQUIRE WEBSERVICE ......................... 437
INQUIRE WORKREQUEST ......................... 440
PERFORM CLASSCACHE ............................ 443
PERFORM CORBASERVER ........................... 447
PERFORM DELETSHIPPED ......................... 450
PERFORM DJAR ..................................... 451
PERFORM DUMP ..................................... 453
PERFORM ENDAFFINITY .......................... 456
PERFORM PIPELINE .............................. 458
PERFORM RESETTIME ............................ 460
PERFORM SECURITY REBUILD ..................... 461
PERFORM SHUTDOWN ............................. 463
PERFORM STATISTICS RECORD .................... 466
RESYNC ENTRYNAME ............................... 471
SET AUTOINSTALL ............................... 474
SET BRFACILITY .................................. 476
SET CLASSCACHE .................................. 478
SET CONNECTION ................................. 480
SET CORBASERVER ............................... 489
SET DB2CONN ..................................... 492
SET DB2ENTRY .................................... 502
SET DB2TRAN ..................................... 508
SET DELETSHIPPED ............................... 510
SET DISPATCHER ................................... 513
SET DSNAME ....................................... 516
SET DUMPDS ....................................... 525
SET ENQMODEL .................................... 527
SET FILE ........................................... 529
SET HOST ........................................... 542
SET IRC ............................................ 544
SET JOURNALNAME .................................. 546
SET JOURNALNUM .................................. 548
ACQUIRE TERMINAL

Acquire a session with a terminal.

**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 VTAM 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 VTAM (the queueing options on the command are for VTAM 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 VTAM request, nor when the terminal is actually acquired, and the terminal does not become associated with the task.

The request is sent straight to VTAM 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 VTAM. For this reason, requests may be queued by VTAM in a different order from the order in which they were originally issued.

**Conditions:** INVREQ, LENGERR, NOTAUTH, TERMIDERR
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 VTAM should not queue the request. Consequently, the ACQUIRE succeeds only if the terminal is immediately available.

QALL
    specifies that VTAM should queue the request if the terminal is not enabled for sessions or is at its session limit (that is, in session with another VTAM application).

QNOTENAB
    specifies that VTAM should queue the request only if the terminal is not enabled for sessions.

QSESSLIM
    specifies that VTAM should queue the request only if the terminal is at its session limit (that is, in session with another VTAM 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 VTAM 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. VTAM simulates a logon when CICS asks to acquire a terminal. This data corresponds to user data that sometimes accompanies a real logon. VTAM 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 LOGON command in [the CICS Application Programming Reference manual] for programming information.

USERDATALEN(data-value)
    specifies the length, as a halfword binary value, of the user data. Because of a VTAM 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-VTAM device.
    4  The terminal is not in service; that is, it is not available for use.
    5  VTAM is not open.
    7  CICS is already in the process of acquiring this session.
ACQUIRE TERMINAL

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.

For statistics relating to the resources URIMAP, PIPELINE, and WEBSERVICE, see "EXTRACT STATISTICS" on page 166.

**COLLECT STATISTICS**

```
COLLECT STATISTICS SET(ptr-ref)
```

**Conditions:** INVREQ, IOERR, NOTAUTH, NOTFND

**last time:**

```
LASTRESET(data-area)
```

**hours:**

```
LASTRESETHRS(data-area)
```

**minutes:**

```
LASTRESETMIN(data-area)
```

**seconds:**

```
LASTRESETSEC(data-area)
```
COLLECT STATISTICS

resource types:

- AUTOINSTALL
- CONNECTION(data-value)
- CORBASERVER(data-value)
- DB2CONN
- DB2ENTRY(data-value)
- DISPATCHER
- ENQUEUE
- FILE(data-value)
- JOURNALNAME(data-value)
- JOURNALNUM(data-value)
- JVMPOOL
- JVMPROFILE(data-value)
- JVMPROGRAM(data-value)
- LSRPOOL(data-value)
- MONITOR
- MVSCTB
- POOL(data-value)
- PROGAUTO
- PROGRAM
- RECOVERY
- REQUESTMODEL(data-value)
- STATS
- STORAGE
- STREAMNAME
- SUBPOOL(data-value)
- SYSDUMPCODE
- TABLEMGR
- TARGET(data-value)
- TASKSUBPOOL
- TCLASS(data-value)
- TCPIP
- TCPIPSERVICE(data-value)
- TOQUEUE
- TERMINAL(data-value)
- TRANCCLASS(data-value)
- TRANDUMPCODE
- TRANSACTION
- TSQUEUE
- VTAM
COLLECT STATISTICS

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).

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 "Programming interface information" on page 734 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 not available for the resource type; similarly, where there is no entry in the specific statistics column, you cannot get statistics for an individual resource.

<table>
<thead>
<tr>
<th>Resource type</th>
<th>Statistic type</th>
<th>Global statistics</th>
<th>Specific statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOINSTALL</td>
<td>Terminal autoinstall</td>
<td>DFHA04DS</td>
<td>-</td>
</tr>
<tr>
<td>BEAN</td>
<td>Enterprise beans</td>
<td>-</td>
<td>DFHEJBDS</td>
</tr>
<tr>
<td>CONNECTION</td>
<td>ISC/IRC system and mode entries</td>
<td>-</td>
<td>DFHA14DS</td>
</tr>
<tr>
<td>CORBASERVER</td>
<td>CorbaServer entries</td>
<td>-</td>
<td>DFHEJRDS</td>
</tr>
<tr>
<td>DB2CONN</td>
<td>DB2 Connection</td>
<td>DFHD2GDS</td>
<td>-</td>
</tr>
<tr>
<td>DB2ENTRY</td>
<td>DB2 Entry</td>
<td>-</td>
<td>DFHD2RDS</td>
</tr>
<tr>
<td>DISPATCHER</td>
<td>Dispatcher</td>
<td>DFHDSGDS</td>
<td>-</td>
</tr>
<tr>
<td>ENQUEUE</td>
<td>Enqueue</td>
<td>DFHNQGDS</td>
<td>-</td>
</tr>
<tr>
<td>FEPI CONNECTION</td>
<td>FEPI Connection</td>
<td>-</td>
<td>DFHA23DS</td>
</tr>
<tr>
<td>FEPI POOL</td>
<td>FEPI Pool</td>
<td>-</td>
<td>DFHA22DS</td>
</tr>
<tr>
<td>FEPI TARGET</td>
<td>FEPI Target</td>
<td>-</td>
<td>DFHA24DS</td>
</tr>
<tr>
<td>FILE</td>
<td>File control</td>
<td>-</td>
<td>DFHA17DS</td>
</tr>
<tr>
<td>JOURNALNAME</td>
<td>Journalname</td>
<td>-</td>
<td>DFHLGRDS</td>
</tr>
<tr>
<td>JOURNALNUM</td>
<td>Journalname</td>
<td>-</td>
<td>DFHLGRDS</td>
</tr>
<tr>
<td>JVMPOOL</td>
<td>JVM pool</td>
<td>DFHSJGDS</td>
<td>-</td>
</tr>
<tr>
<td>JVMPROFILE</td>
<td>JVM profiles</td>
<td>-</td>
<td>DFHSJRDS</td>
</tr>
<tr>
<td>JVMPROGRAM</td>
<td>JVM programs</td>
<td>-</td>
<td>DFHPGRDS</td>
</tr>
</tbody>
</table>
**COLLECT STATISTICS**

Table 1. Resource types and statistics (continued)

<table>
<thead>
<tr>
<th>Resource type</th>
<th>Statistic type</th>
<th>Global statistics</th>
<th>Specific statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSRPOOL</td>
<td>LSR pools</td>
<td>-</td>
<td>DFHA08DS</td>
</tr>
<tr>
<td>MONITOR</td>
<td>Monitor</td>
<td>DFHMNGDS</td>
<td>DFHMNTDS</td>
</tr>
<tr>
<td>MVSTCB</td>
<td>MVS TCB</td>
<td>DFHDSTDS</td>
<td>DFHDSRDS</td>
</tr>
<tr>
<td>PROGAUTO</td>
<td>Program autoinstall</td>
<td>DFHPGGDS</td>
<td>-</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>Program</td>
<td>DFHLDGDS</td>
<td>DFHLDRDS</td>
</tr>
<tr>
<td>RECOVERY</td>
<td>Recovery manager</td>
<td>DFHRMGDS</td>
<td>-</td>
</tr>
<tr>
<td>REQUESTMODEL</td>
<td>Request models</td>
<td>-</td>
<td>DFHIIRDS</td>
</tr>
<tr>
<td>STATS</td>
<td>Statistics</td>
<td>DFHSTGDS</td>
<td>-</td>
</tr>
<tr>
<td>STORAGE</td>
<td>Storage manager (SM)</td>
<td>DFHSMSDS</td>
<td>DFHSMDDS</td>
</tr>
<tr>
<td>STREAMNAME</td>
<td>Log stream</td>
<td>DFHLGGDS</td>
<td>DFHLGSDS</td>
</tr>
<tr>
<td>SUBPOOL</td>
<td>SM domain subpool</td>
<td>-</td>
<td>DFHSMDDS</td>
</tr>
<tr>
<td>SYSDUMPCODE</td>
<td>Dump (system)</td>
<td>DFHSDGDS</td>
<td>DFHSDRDS</td>
</tr>
<tr>
<td>TABLEMGR</td>
<td>Table manager</td>
<td>DFHA16DS</td>
<td>-</td>
</tr>
<tr>
<td>TASKSUBPOOL</td>
<td>SM task subpool</td>
<td>DFHSMTDS</td>
<td>-</td>
</tr>
<tr>
<td>TCLASS</td>
<td>Transaction class</td>
<td>-</td>
<td>DFHXMCDS</td>
</tr>
<tr>
<td>TCPIP</td>
<td>Sockets domain</td>
<td>DFHSOGDS</td>
<td>-</td>
</tr>
<tr>
<td>TCPIPIService</td>
<td>TCP/IP service</td>
<td>-</td>
<td>DFHSORDS</td>
</tr>
<tr>
<td>TDQUEUE</td>
<td>Transient data</td>
<td>DFHTQGDS</td>
<td>DFHTQRDS</td>
</tr>
<tr>
<td>TERMINAL</td>
<td>Terminals</td>
<td>-</td>
<td>DFHA06DS</td>
</tr>
<tr>
<td>TRANCLASS</td>
<td>Transaction class</td>
<td>-</td>
<td>DFHXMCDS</td>
</tr>
<tr>
<td>TRANDUMPCODE</td>
<td>Dump (transaction)</td>
<td>DFHTDGDS</td>
<td>DFHTDRDS</td>
</tr>
<tr>
<td>TRANSACTION</td>
<td>Transaction manager</td>
<td>DFHXMGDS</td>
<td>DFHXMRDS</td>
</tr>
<tr>
<td>TSQUEUE</td>
<td>Temporary storage</td>
<td>DFHTSGDS</td>
<td>-</td>
</tr>
<tr>
<td>VTAM</td>
<td>VTAM</td>
<td>DFHA03DS</td>
<td>-</td>
</tr>
</tbody>
</table>

Copybooks are provided in ASSEMBLER, COBOL, and PL/I. (There is no copybook for C.) The names of the copybooks are the same in each language. You can find them in the following libraries:

- **ASSEMBLER**: CICSTS31.CICS.SDFHMAC
- **COBOL**: CICSTS31.CICS.SDFHCOB
- **PL/I**: CICSTS31.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](https://example.com).

**Options**

**AUTOINSTALL**  
requests global statistics on autoinstall.
COLLECT STATISTICS

**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 may 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 DFHnn, where nn is the journal number in the range 01 to 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 half-word 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 releases of CICS earlier than CICS Transaction Server for z/OS. However, the statistics returned are CICS log manager statistics, not journal control.
COLLECT STATISTICS

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, you are recommended 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 HFS 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).

**LSRPPOOL**(*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. (For programming information, see EIBTASKN in Appendix A of the CICS Application Programming Reference manual) Without a data-value, MONITOR requests global performance class statistics.
COLLECT 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 on 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 need to 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.

TRANCLASS\((data-value)\)
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,
TRANCLASS 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
requests global statistics on VTAM.
COLLECT STATISTICS

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 makes use of virtually all the options described in this section. This set of programs illustrates ways of using the COLLECT STATISTICS and INQUIRE commands of CICS Transaction Server for z/OS 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 on installing and operating the DFH0STAT application. The source code for the application can be found in CICSTS31.CICS.SDFHSAMP.

Restrictions

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 CONNECTION

Define a CONNECTION in the local CICS region.

CREATE CONNECTION

CREATE CONNECTION (data-value)  ATTRIBUTES (data-value)  ATTRLEN (data-value)
  LOGMESSAGE (cvda)
  COMPLETE
  DISCARD

CREATE CONNECTION attribute values:

- **ACCESSMETHOD**: VTAM, INDIRECT, IRC, XM
- **ATTACHSEC**: LOCAL, IDENTITY, MIXIDPE, PERSISTENT, VERIFY
- **AUTOCONNECT**: NO, ALL, YES
- **BINDSECURITY**: NO, YES
- **CONNTYPE**: SPECIFIC, GENERIC
- **DATASTREAM**: USER, LMS, SCS, STRFIELD, 3270
- **DESCRIPTION**: CHAR58
- **INDSYS**: CHAR4
- **INSERVICE**: YES, NO
- **MAXQTIME**: NO, 0-9999
- **NETNAME**: CHAR8
- **PROTOCOL**: APPC, EXCI, LU61
- **PSRECOVERY**: SYSDEFAULT, NONE
- **QUEUELIMIT**: NO, 0-9999
- **RECORDFORMAT**: U, VB
- **REMOTESYSNET**: CHAR8
- **REMOTESYSTEM**: CHAR4
- **REMTENAME**: CHAR4
- **SINGLESESS**: NO, YES
- **USEDFLTUSER**: NO, YES
- **XLNACTION**: KEEP, FORCE

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

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.

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 "ATTRIBUTES option" on page 26 for general
CREATE CONNECTION

rules for specifying attributes, and the CONNECTION chapter in the CICS Resource Definition Guide 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**
specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

- **LOG** The resources attributes are logged to the CSDL transient data queue.
- **NOLOG** The resources attributes are not logged to the CSDL transient data queue.

**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, “EXEC CICS CREATE RESP2 values,” on page 681 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.
CREATE CONNECTION

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**

```
CREATE CORBASERVER(data-value) ATTRIBUTES(data-value) ATTRLEN(data-value) LOGMESSAGE(cvda)
```

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

**CREATE CORBASERVER attributes:**

- **DESCRIPTION(char58)**
- **DJARDIR(char255)**
- **STATUS(ENABLED)**
- **STATUS(DISABLED)**
- **AUTOPUBLISH(NO)**
- **AUTOPUBLISH(YES)**
- **JNDIPREFIX(char255)**
- **SESSBEANTIME(00,00,10)**
- **SESSBEANTIME(int2, int2, int2)**
- **SHELF(/var/cicsts)**
- **SHELF(char255)**
- **HOST(char255)**
- **CERTIFICATE(char56)**
- **UNAUTH(char8)**
  - **ASSERTED(char8)**
  - **CLIENTCERT(char8)**
  - **SSLUNAUTH(char8)**
  - **CIPHERS(data-area)**
- **OUTPRIVACY(SUPPORTED)**
- **OUTPRIVACY(NOTSUPPORTED)**
- **OUTPRIVACY(REQUIRED)**

**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](#) 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 [ATTRIBUTES option](#) for general
CREATE CORBASERVER

rules for specifying attributes, and the CORBASERVER chapter in 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
specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

LOG The resources attributes are logged to the CSDL transient data queue.

NOLOG The resources attributes are not logged to the CSDL transient data queue.

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 which identifies more precisely the nature of the error. See Appendix C, “EXEC CICS CREATE RESP2 values,” on page 681 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 JNDI prefix, 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 one of its associated TCPIPSERVICES was not installed.

LENGERR
RESP2 values:
1 The length specified in ATTRLEN is negative.

NOTAUTH
RESP2 values:
CREATE CORBASERVER

   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(data-value) -- ATTRIBUTES(data-value) -- ATTRLEN(data-value) -- LOGMESSAGE(cvda)

CREATE DB2CONN attribute values:

DESCRIPTION(char58) -- CONNECTERROR(sqlcode)

CONNECTERROR(abend) -- DB2GROUPID(char4) -- DB2ID(char4)

MSGQUEUE1(CDB2) -- MSGQUEUE2(char4) -- MSGQUEUE3(char4)

PURGECYCLE(0,30) -- RESYNCMEMBER(yes) -- STANDBYMODE(reconnect)

PURGECYCLE(0-59,0-59) -- RESYNCMEMBER(no) -- STANDBYMODE(noconnect)

NONTERMREL(yes) -- NONTERMREL(no)

STATSQUEUE(CDB2) -- STATSQUEUE(char4)

TCBLIMIT(12) -- THREADERROR(n906d)

TCBLIMIT(4-2000) -- THREADERROR(abend) -- THREADERROR(n906)

ACCOUNTREC(accnone) -- ACCOUNTREC(uow) -- ACCOUNTREC(task) -- ACCOUNTREC(txid)

AUTHID(char8) -- AUTHTYPE(userid)

AUTHTYPE(group) -- DROLLBACK(yes)

AUTHTYPE(signid) -- DROLLBACK(no)

AUTHTYPE(term) -- PLAN(char8)

AUTHTYPE(opid)

PLANEXITNAME(DSNCUEXT) -- PRIORITY(high) -- THREADLIMIT(3)

PLANEXITNAME(char8) -- PRIORITY(equal) -- THREADLIMIT(3-2000) -- THREADWAIT(yes)

PRIORITY(low) -- THREADLIMIT(3)

COMAUTHID(char8) -- COMAUTHTYPE(userid)

COMAUTHTYPE(group) -- COMTHREADLIM(1)

COMAUTHTYPE(signid)

COMAUTHTYPE(term)

COMAUTHTYPE(txid)

COMAUTHTYPE(user)

Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH
CREATE DB2CONN

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 syncpoint 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 being added. The list of attributes must be coded as a single character string using the syntax shown in DB2CONN attributes. See [ATTRIBUTES option on page 26](#) for general rules for specifying attributes, and the DB2CONN chapter in 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**

specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

- **LOG** The resources attributes are logged to the CSDL transient data queue.
- **NOLOG** The resources attributes are not logged to the CSDL transient data queue.

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, “EXEC CICS CREATE RESP2 values,” on page 681 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 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) — ATTRIBUTES(data-value) — ATTRLEN(data-value) — LOGMESSAGE(cvda)

CREATE DB2ENTRY attribute values:

- DESCRIPTION(char58)
- TRANSID(char4)
- ACCOUNTREC(uow)
- ACCOUNTREC(task)
- ACCOUNTREC(txid)
- AUTHID(char8)
- AUTHTYPE(userid)
- AUTHTYPE(group)
- AUTHTYPE(signid)
- AUTHTYPE(term)
- AUTHTYPE(txid)
- AUTHTYPE(opid)
- DROLLBACK(yes)
- DROLLBACK(no)
- PLAN(char8)
- PLANEXITNAME(DSNCUEXT)
- PLANEXITNAME(char8)
- PLANEXITNAME(pool)
- PROTECTNUM(0)
- THREADLIMIT(0)
- THREADWAIT(pool)
- THREADWAIT(yes)
- THREADWAIT(no)

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.
CREATE DB2ENTRY

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 [“ATTRIBUTES option” on page 26](#) for general rules for specifying attributes, and the DB2ENTRY chapter in the CICS 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**

specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

- **LOG** The resources attributes are logged to the CSDL transient data queue.
- **NOLOG** The resources attributes are not logged to the CSDL transient data queue.

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, “EXEC CICS CREATE RESP2 values,” on page 681 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.
CREATE DB2ENTRY

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 (data-value) ATTRIBUTES (data-value) ATTRLEN (data-value) LOGMESSAGE (cvda)

CREATE DB2TRAN attribute values:

- DESCRIPTION (char58)
- ENTRY (char8)
- TRANSID (char4)

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 "ATTRIBUTES option" on page 26 for general rules for specifying attributes, and the DB2TRAN chapter in 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.
**CREATE DB2TRAN**

**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**
specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

- **LOG** The resources attributes are logged to the CSDL transient data queue.
- **NOLOG** The resources attributes are not logged to the CSDL transient data queue.

**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, “EXEC CICS CREATE RESP2 values,” on page 681](#) 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.

**LENERR**
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

\[\text{\textbackslash CREATE DJAR(data-value)---ATTRIBUTES(data-value)---ATTRLEN(data-value)}\]

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

CREATE DJAR attributes:

\[\text{DESCRIPTION(char58)---CORBASERVER(char4)---HFSFILE(char255)}\]

**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 "ATTRIBUTES option" on page 26 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.
CREATE DJAR

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, “EXEC CICS CREATE RESP2 values,” on page 681 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

\[\text{CREATE DOCTEMPLATE}(\text{data-value})\rightarrow\text{ATTRIBUTES}(\text{data-value})\rightarrow\text{ATTRLEN}(\text{data-value})\rightarrow\text{LOGMESSAGE}(\text{cvda})\rightarrow\]

CREATE DOCTEMPLATE attribute values:

- APPENDCRLF(YES)
- APPENDCRLF(NO)
- DESCRIPTION(char58)
- FILE(char8)
- TSQUEUE(char16)
- TQUEUE(char4)
- PROGRAM(char8)
- EXITPGM(char8)
- TEMPLATENAME(char48)
- HFSFILE(char255)
- DDNAME(DFHHTML)
- ODNAME(char8)

Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

Note to COBOL programmers:

In the syntax above, you must use

\[\text{--- ATTRIBUTES (data-area)}\rightarrow\text{--- ATTRIBUTES (data-value)}\rightarrow\]

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

\text{ATTRIBUTES}(\text{data-value})

specifies the attributes of the DOCTEMPLATE being added. The list of
CREATE DOCTEMPLATE

attributes must be coded as a single character string using the syntax shown in
DOCTEMPLATE attributes. See "ATTRIBUTES option" on page 26 for general
rules for specifying attributes, and the DOCTEMPLATE chapter in 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.

**DOCTEMPLATE(data-value)**
specifies the 8-character name of the DOCTEMPLATE definition to be added to
the CICS region.

**LOGMESSAGE**
specifies whether CICS logs the attributes used for the resource being created
to the CSDL transient data queue. CVDA values are:

- **LOG** The resources attributes are logged to the CSDL transient data
  queue.
- **NOLOG** The resources attributes are not logged to the CSDL transient
data queue.

**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, “EXEC CICS CREATE RESP2 values,” on page 681
  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.
CREATE ENQMODEL

Define an ENQMODEL resource definition.

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 an 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.

Conditions: INVREQ, LENGERR, NOTAUTH
CREATE ENQMODEL

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 **ENQMODEL attributes**. See ["ATTRIBUTES option" on page 26](#) for general rules for specifying attributes, and the ENQMODEL chapter in 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.

**ENQMODEL**(data-value)
specifies the 8-character name of the ENQMODEL definition to be added to the CICS region.

**LOGMESSAGE**
specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

- **LOG** The resources attributes are logged to the CSDL transient data queue.
- **NOLOG** the resources attributes are not logged to the CSDL transient data queue.

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, “EXEC CICS CREATE RESP2 values,” on page 681](#) 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.

**LENERR**
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 an ENQMODEL definition with this name.
CREATE FILE

Define a FILE in the local CICS region.

CREATE FILE

CREATE FILE(data-value)—ATTRIBUTES(data-value)—ATTRLEN(data-value)—LOGMESSAGE(cvda)

Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

CREATE FILE attribute values (part 1):

ADD(NO) — BACKUPTYPE STATIC) — BROWSE(NO) — CFDTPOOL(char8)
ADD(YES) — BACKUPTYPE (DYNAMIC) — BROWSE(YES)

DATABUFFERS(2) — DELETE(NO) — DISPOSITION(SHARE)
DATABUFFERS(2-32767) — DELETE(YES) — DISPOSITION(OLD)

DSNAME(char44) — DSNSHARING(ALLREQS) — FWDCOVLOG(NO)
DSNSHARING(MODIFYREQS) — FWDCOVLOG(1-99)

INDEXBUFFERS(1) — JNLADD(NONE) — JNLREAD(NONE) — JNLSYSCREAD(NO)
INDEXBUFFERS(1-32767) — JNLADD(AFTER) — JNLREAD(ALL) — JNLSYSCREAD(YES)
INDEXBUFFERS(1-32767) — JNLADD(BEFORE) — JNLREAD(READONLY) — JNLSYSCREAD(YES)
INDEXBUFFERS(1-32767) — JNLADD(BEFORE) — JNLREAD(UPDATEONLY) — JNLSYSCREAD(YES)

JNLUPDATE(NO) — JNLUPDATE(YES)
JNLSYNCWRITE(YES) — JNLSYNCWRITE(NO)

KEYLENGTH(1-255|1-16 for CFDT) — LOAD(NO) — LSRPOOLID(1)
KEYLENGTH(1-255|1-16 for CFDT) — LOAD(YES) — LSRPOOLID(NONE)
KEYLENGTH(1-255|1-16 for CFDT) — LOAD(YES) — LSRPOOLID(I-8)

MAXNUMRECS(NOLIMIT) — NSRGROUP(char8) — OPENTIME(FIRSTREF)
MAXNUMRECS(1-99999999) — NSRGROUP(char8) — OPENTIME(STARTUP)

READ(NO) — READINTEG(UNCOMMITTED) — RECORDFORMAT(V)
READ(NO) — READINTEG(CONSISTENT) — RECORDFORMAT(F)
READ(NO) — READINTEG(REPEATABLE) — RECORDFORMAT(F)
CREATE FILE

CREATE FILE attributes (part 2):

- RECORDSIZE(1-32767)
- RECOVERY(NONE)
- RECOVERY(ALL)
- RECOVERY(BACKOUTONLY)
- REMOTESYSTEM(char4)
- REMOTENAME(char8)
- RLSACCESS(NO)
- RLSACCESS(YES)
- STATUS(ENABLED)
- STATUS(DISABLED)
- STATUS(UNENABLED)
- STRINGS(1)
- STRINGS(1-255)
- TABLE(NO)
- TABLE(CF)
- TABLE(CICS)
- TABLE(USER)
- TABLENAMESPACE(char8)
- TABLENAME(char8)
- UPDATE(NO)
- UPDATE(YES)
- UPDATEMODEL(LOCKING)
- UPDATEMODEL(CONTENTION)

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 syncpoint 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 being added. The list of attributes must be coded as a single character string using the syntax shown in FILE attributes. See "ATTRIBUTES option" on page 26 for general rules for specifying attributes, and the FILE chapter in 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 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.

FILE(data-value)
specifies the 8-character name of the FILE definition to be added to the CICS region.

LOGMESSAGE
specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

LOG The resources attributes are logged to the CSDL transient data queue.

NOLOG The resources attributes are not logged to the CSDL transient data queue.

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, “EXEC CICS CREATE RESP2 values,” on page 681 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 FILE definition with this name.
CREATE JOURNALMODEL

Define a journal model in the local CICS region.

CREATE JOURNALMODEL

CREATE JOURNALMODEL

CREATE JOURNALMODEL attribute values:

- DESCRIPTION(char58)
- JOURNALNAME(char8)
- STREAMNAME(&USERID..&APPLID..&JNAME;)
- STREAMNAME(char26)
- TYPE(MVS)
- TYPE(DUMMY)
- TYPE(SMF)

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; it 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 "ATTRIBUTES option" on page 26 for general rules for specifying attributes, and the JOURNALMODEL chapter in the CICS Resource Definition Guide for details about specific attributes.
CREATE JOURNALMODEL

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
specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

LOG The resources attributes are logged to the CSDL transient data queue.
NOLOG The resources attributes are not logged to the CSDL transient data queue.

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, “EXEC CICS CREATE RESP2 values,” on page 681 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 LSRPOOL

 Define an LSR pool in the local CICS region.

CREATE LSRPOOL 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 LSRPOOL command installs a LSRPOOL definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. LSR pools must have unique LSRPOOLID values within a CICS region. If the local region already contains a definition with the same LSRPOOLID 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 LSRPOOLID value governs.)

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 using it are closed and one is reopened
subsequently.

A syncpoint 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 being added. The list of attributes must
be coded as a single character string using the syntax shown in LSRPOOL
attributes. See "ATTRIBUTES option" on page 26 for general rules for
specifying attributes, and the LSRPOOL chapter in the CICS Resource
Definition Guide for details about specific attributes.

Note: You can assign default values for all attributes of a 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
specifies whether CICS logs the attributes used for the resource being created
to the CSDL transient data queue. CVDA values are:

LOG The resources attributes are logged to the CSDL transient data
queue.

NOLOG The resources attributes are not logged to the CSDL transient
data queue.

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, "EXEC CICS CREATE RESP2 values," on page 681
for information on RESP2 values.
7 The LOGMESSAGE cvda value is not valid.
CREATE LSRPOOL

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(data-value) — ATTRIBUTES(data-value) — ATTRLEN(data-value) — LOGMESSAGE(cvda)
```

Create MAPSET attribute values:

- DESCRIPTION(char58)
- RESIDENT(YES) — STATUS(ENABLED) — USAGE(NORMAL)
- RESIDENT(NO) — STATUS(DISABLED) — USAGE(TRANSIENT)
- USELPACOPY(NO)
- USELPACOPY(YES)

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 "ATTRIBUTES option" on page 26 for general rules for specifying attributes, and the MAPSET chapter in the CICS Resource Definition Guide for details about specific attributes.
CREATE MAPSET

**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 32767.

**LOGMESSAGE**

specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

- **LOG**
  The resources attributes are logged to the CSDL transient data queue.

- **NOLOG**
  The resources attributes are not logged to the CSDL transient data queue.

**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, “EXEC CICS CREATE RESP2 values,” on page 681](#) 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 PARTITIONSET

Define a partition set in the local CICS region.

CREATE PARTITIONSET

- CREATE PARTITIONSET(data-value)—ATTRIBUTES(data-value)—ATTRLEN(data-value)
- LOGMESSAGE(cvda)

CREATE PARTITIONSET attribute values:

- DESCRIPTION(char58)
- RESIDENT(NO)
- STATUS(ENABLED)
- USAGE(NORMAL)
- USELPACOPY(NO)
- RESIDENT(YES)
- STATUS(DISABLED)
- USAGE(TRANSIENT)
- USELPACOPY(YES)

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 synchpoint 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 "ATTRIBUTES option" on page 26 for general rules for specifying attributes, and the PARTITIONSET chapter in the CICS Resource Definition Guide for details about specific attributes.
CREATE PARTITIONSET

**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 32767.

**LOGMESSAGE**

specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

- **LOG** The resources attributes are logged to the CSDL transient data queue.
- **NOLOG** The resources attributes are not logged to the CSDL transient data queue.

**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, “EXEC CICS CREATE RESP2 values,” on page 681](#) 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 DLPLSUBSET 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

\[\text{CREATE PARTNER(data-value)---ATTRIBUTES(data-value)---ATTRLEN(data-value)---LOGMESSAGE(cvda)}\]

**CREATE PARTNER attribute values:**

- DESCRIPTION(char58)
- NETNAME(char8)
- NETWORK(char8)
- PROFILE(DFHCICSA)
- TPNAME(char64)
- XTPNAME(hex128)

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

---

**Note to COBOL programmers:**

In the syntax above, you must use

\[---\text{ATTRIBUTES( data-area )}---\]

instead of

\[---\text{ATTRIBUTES( data-value)}---\]

---

**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 "ATTRIBUTES option" on page 26 for general rules for specifying attributes, and the PARTNER chapter in 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.
CREATE PARTNER

LOGMESSAGE

specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

LOG The resources attributes are logged to the CSDL transient data queue.

NOLOG The resources attributes are not logged to the CSDL transient data queue.

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, “EXEC CICS CREATE RESP2 values,” on page 681 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(data-value)—ATTRIBUTES(data-value)—ATTRLEN(data-value)—LOGMESSAGE(cvda)

CREATE PIPELINE attribute values:

- DESCRIPTION(text)
- CONFIGFILE(name)
- WSDIR(directory)
- SHELF(/var/cicsts)
- SHELF(directory)
- STATUS(ENABLED)
- STATUS(DISABLED)

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 "ATTRIBUTES option" on page 26 for general rules for specifying attributes, and the PIPELINE chapter in the CICS Resource Definition Guide for details about specific attributes.
CREATE PIPELINE

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.

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, “EXEC CICS CREATE RESP2 values,” on page 681 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(data-value) --- ATTRIBUTES(data-value) --- ATTRLEN(data-value) --- LOGMESSAGE(cvda) ---

CREATE PROCESSTYPE attribute values:

- DESCRIPTION(char58)
- FILE(char8)
- AUDITLOG(char8)
- AUDITLEVEL(OFF)
- AUDITLEVEL(ACTIVITY)
- AUDITLEVEL(FULL)
- AUDITLEVEL(PROCESS)

- STATUS(ENABLED)
- STATUS(DISABLED)

Conditions: ILLOGIC, INVREQ, LENERR, 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 "ATTRIBUTES option" on page 26 for general rules for specifying attributes, and the PROCESSTYPE chapter in the CICS Resource Definition Guide for details about specific attributes.
CREATE PROCESSTYPE

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
specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

LOG The resources attributes are logged to the CSDL transient data queue.
NOLOG The resources attributes are not logged to the CSDL transient data queue.

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, “EXEC CICS CREATE RESP2 values,” on page 681 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(data-value)---ATTRIBUTES(data-value)---ATTRLEN(data-value)---LOGMESSAGE(cvda)
```

CREATE PROFILE attribute values:

- CHAINCONTROL(NO)
- CHAINCONTROL(YES)
- DESCRIPTION(char58)
- DVSUPRT(ALL)
- DVSUPRT(NONVTAM)
- DVSUPRT(VTAM)
- INBFMH(NO)
- INBFMH(ALL)
- INBFMH(DIP)
- INBFMH(EODS)
- JOURNAL(NO)
- JOURNAL(1-99)
- LOGREC(NO)
- LOGREC(YES)
- MODENAME(char8)
- MSGINTEG(NO)
- MSGINTEG(YES)
- MSGJRNL(NO)
- MSGJRNL(INOUT)
- MSGJRNL(INPUT)
- MSGJRNL(OUTPUT)
- NEPCLASS(0)
- NEPCLASS(0-255)
- ONEWTE(NO)
- ONEWTE(YES)
- PRINTERCOMP(NO)
- PRINTERCOMP(YES)
- PROTECT(NO)
- PROTECT(YES)
- RAQ(NO)
- RAQ(YES)
- RTIMOUT(NO)
- RTIMOUT(1-7000)
- SCRNSIZE(DEFAULT)
- SCRNSIZE(ALTERNATE)
- UCTRAN(NO)
- UCTRAN(YES)

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 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.
CREATE PROFILE

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 "ATTRIBUTES option" on page 26 for general rules for specifying attributes, and the PROFILE chapter in 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 32767.

LOGMESSAGE
specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

- LOG The resources attributes are logged to the CSDL transient data queue.
- NOLOG The resources attributes are not logged to the CSDL transient data queue.

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, “EXEC CICS CREATE RESP2 values,” on page 681 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 attribute values:

CEDF(YES) — CEDF(NO)
CONCURRENCY(QUASIRENT) — CONCURRENCY(THREADSAFE)
DATolocation(BELOW) — DATolocation(ANY)
DESCRIPTION(char58)
EXECUTIONSET(FULLAPI) — EXECUTIONSET(DPSUBSET)
JVMCLASS(char255) — JVM(NO) — JVM(YES)
RELOAD(NO) — RELOAD(YES)
JVMPROFILE(char8)
LANGUAGE(ASSEMBLER) — LANGUAGE(C) — LANGUAGE(COBOL) — LANGUAGE(LE370) — LANGUAGE(PLI)
RESIDENT(NO) — RESIDENT(YES)
STATUS(ENABLED) — STATUS(DISABLED)
TRANSID(char4) — USALGE(NORMAL) — USELPACOPY(NO)
USAGE(NORMAL) — USAGE(TRANSIENT) — USELPACOPY(YES)

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.
CREATE PROGRAM

A syncpoint 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 "ATTRIBUTES option" on page 26 for general rules for specifying attributes, and the PROGRAM chapter in the CICS 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 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
Specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

LOG The resources attributes are logged to the CSDL transient data queue.

NOLOG The resources attributes are not logged to the CSDL transient data queue.

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, “EXEC CICS CREATE RESP2 values,” on page 681 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.
CREATE 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 create a PROGRAM definition with this name.
CREATE REQUESTMODEL

Define a request model in the local CICS region.

<table>
<thead>
<tr>
<th>CREATE REQUESTMODEL attribute values:</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESCRIPTION(char58)</td>
</tr>
<tr>
<td>TRANSID(char4)</td>
</tr>
<tr>
<td>CORBASERVER(char4)</td>
</tr>
<tr>
<td>TYPE(GENERIC)</td>
</tr>
<tr>
<td>TYPE(EJB)</td>
</tr>
<tr>
<td>TYPE(CORBA)</td>
</tr>
<tr>
<td>BEANNAME(char240)</td>
</tr>
<tr>
<td>INTFACETYPE(BOTH)</td>
</tr>
<tr>
<td>INTFACETYPE(HOME)</td>
</tr>
<tr>
<td>INTFACETYPE(REMOTE)</td>
</tr>
<tr>
<td>INTERFACE(char255)</td>
</tr>
<tr>
<td>MODULE(char255)</td>
</tr>
<tr>
<td>OPERATION(char255)</td>
</tr>
</tbody>
</table>

Conditions: ILLOG, 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; it 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.
CREATE REQUESTMODEL

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 "ATTRIBUTES option" on page 26 for general rules for specifying attributes, and the REQUESTMODEL chapter in 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
specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

LOG The resources attributes are logged to the CSDL transient data queue.

NOLOG The resources attributes are not logged to the CSDL transient data queue.

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, "EXEC CICS CREATE RESP2 values," on page 681 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

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
CREATE SESSIONS

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 48 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 "ATTRIBUTES option" on page 26 for general rules for specifying attributes, and the SESSIONS chapter in 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**

specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

- **LOG** The resources attributes are logged to the CSDL transient data queue.
- **NOLOG** The resources attributes are not logged to the CSDL transient data queue.

**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, "EXEC CICS CREATE RESP2 values,” on page 681 for information on RESP2 values.
- **7** The LOGMESSAGE cvda value is not valid.
- **200** The command was executed in a program defined with an
CREATE SESSIONS

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 TCPIPSERVICE

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

CREATE TCPIPSERVICE

CREATE TCPIPSERVICE(data-value) — ATTRIBUTES(data-value) — ATTRLN(data-value)

LOGMESSAGE(cvda)  PROTOCOL(ECI):  PROTOCOL(ECI)  ATTACHSEC(VERIFY)

CREATE TCPIPSERVICE attribute values:

BACKLOG(0)  BACKLOG(1-32767)  CERTIFICATE(char44)  DESCRIPTION(char58)  DNSGROUP(name)

GRPCRITICAL(NO)  GRPCRITICAL(YES)  IPADDRESS(char15)  MAXDATALEN(3-536870)  PORTNUMBER(1-65535)

PRIVACY(SUPPORTED)  PRIVACY(NOT_SUPPORTED)  PRIVACY(REQUIRED)  PROTOCOL(HTTP)  SSL(NO)

PROTOCOL(USER)  PROTOCOL(IIOP)  SSL(YES)  CIPHERS(data-area)

PROTOCOL(EI)  SSL(CLIENTAUTH)

STATUS(OPEN)  STATUS(CLOSED)  TRANSACTION(CWXN)  TRANSACTION(char4)  TSQPREFIX(char6)

PROTOCOL(HTTP) or PROTOCOL(USER):

PROTOCOL(HTTP)  AUTHENTICATE(NO)  AUTHENTICATE(AUTOMATIC)

AUTHENTICATE(AUTOREGISTER)  AUTHENTICATE(BASIC)  AUTHENTICATE(CERTIFICATE)  AUTHENTICATE(ASSERTED)

PROTOCOL(IIOP):

PROTOCOL(IIOP)  AUTHENTICATE(NO)  AUTHENTICATE(CERTIFICATE)

AUTHENTICATE(ASSERTED)

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 TCPIPSERVICE command installs a TCPIPSERVICE definition with
the attribute specified on the command. It does not use a resource definition stored
in the CSD. If there is already a TCP/IP service by 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 TCPIPSERVICE 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 TCPIPSERVICE being added. The list of attributes
must be coded as a single character string using the syntax shown in
CONNECTION attributes. See “ATTRIBUTES option” on page 26 for general
rules for specifying attributes, and the TCPIPSERVICE chapter in 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
specifies whether CICS logs the attributes used for the resource being created
to the CSDL transient data queue. CVDA values are:

LOG The resources attributes are logged to the CSDL transient data
queue.

NOLOG The resources attributes are not logged to the CSDL transient
data queue.

TCPIPSERVICE(data-value)
specifies the 8-character name of the TCPIPSERVICE definition to be added to
the CICS region.

Conditions

ILLLOGIC
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, “EXEC CICS CREATE RESP2 values,” on page 681 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 TDQUEUE

Define a transient data queue in the local CICS region.

CREATE TDQUEUE

CREATE TDQUEUE(data-value)—ATTRIBUTES(data-value)—ATTRLEN(data-value)—LOGMESSAGE(cvda)

Conditions: ILLOGIC, INVREQ, LENGERR, NOTAUTH

CREATE TDQUEUE attribute values for extra-partition queues:

- TYPE(EXTRA)
- BLOCKFORMAT(BLOCKED)
- BLOCKFORMAT(UNBLOCKED)
- BLOCKSIZE(0)
- BLOCKSIZE(0-32767)
- DATABUFFERS(1)
- DATABUFFERS(1-255)
- DDNAME(char8)
- DESCRIPTION(char58)
- DISPOSITION(SHR)
- DISPOSITION(OLD)
- DISPOSITION(MOD)
- DSNAME(DUMMY)
- DSNAME(char44)
- ERROROPTION(IGNORE)
- ERROROPTION(SKIP)
- OPENTIME(INITIAL)
- OPENTIME(DEFERRED)
- PRINTCONTROL(A)
- PRINTCONTROL(M)
- RECORDSIZE(0)
- RECORDSIZE(0-32767)
- RECORDFORMAT(FIXED)
- RECORDFORMAT(VARIABLE)
- SYSOUTCLASS(*)
- SYSOUTCLASS(char1)
- TYPEFILE(INPUT)
- TYPEFILE(OUTPUT)
- TYPEFILE(RDBACK)
- Remote attributes
- Rewind(LEAVE)
- Rewind(REREAD)
CREATE TDQUEUE attribute values for intra-partition queues:

- TYPE(INTRA)
- ATIFACILITY(TERMAL)
- ATIFACILITY(FILE)
- ATIFACILITY(SYSTEM)
- DESCRIPTION(char58)
- FACILITYID(char4)
- RECOVSTATUS(NO)
- RECOVSTATUS(LOGICAL)
- RECOVSTATUS(PHYSICAL)
- TRIGGERLEVEL(1)
- TRIGGERLEVEL(0-32767)
- TRANSID(char4)
- USERID(char8)
- WAIT(YES)
- WAITACTION(REJECT)
- WAIT(NO)
- WAITACTION(QUEUE)

CREATE TDQUEUE attribute values for indirect queues:

- TYPE(INDIRECT)
- DESCRIPTION(char58)
- INDIRECTNAME(char4)
- Remote attributes

CREATE TDQUEUE attribute values for remote queues of unspecified TYPE:

- DESCRIPTION(char58)
- Remote attributes

Remote attributes:

- REMOTESYSTEM(char4)
- REMOTENAME(tdqueue)
- REMOTENAME(char4)
- REMOTELENGTH(0-32767)

Note to COBOL programmers:

In the syntax above, you must use
--- ATTRIBUTES (data-area)--- instead of
--- ATTRIBUTES (data-value)---
CREATE TDQUEUE

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 "ATTRIBUTES option" on page 26 for general rules for specifying attributes, and the TDQUEUE chapter in 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

specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

LOG

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

NOLOG

The resources attributes are not logged to the CSDL transient data queue.

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, “EXEC CICS CREATE RESP2 values,” on page 681 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.

**LENERR**

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(data-value) ATTRIBUTES(data-value) ATTRLEN(data-value)
```

CREATE TERMINAL attribute values:

```
ALTPRINTCOPY(NO) ALTPRINTER(char4) ATTACHSEC(Local)
```

```
ALTPRINTCOPY(YES) ATTACHSEC(IDENTIFY) ATTACHSEC(MIXIDPE)
```

```
ATTACHSEC(PERSISTENT) ATTACHSEC(VERIFY)
```

```
AUTINSTMODEL(NO) AUTINSTNAME(char8) BINDPASSWORD(hex16)
```

```
AUTINSTMODEL(ONLY) AUTINSTMODEL(YES) BINDSECURITY(NO)
```

```
AUTINSTMODEL(YES) BINDSECURITY(YES)
```

```
CONSOLE(NO) CONSNAMEN(char8) DESCRIPTION(char58)
```

```
INSERVICE(YES) INSERVICE(NO) MODENAME(char8)
```

```
AUTINSTMODEL(YES) PRINTERCOPY(NO)
```

```
AUTINSTMODEL(YES) PRINTERCOPY(YES)
```

```
NATLANG(E) NETNAME(char8) POOL(char8) PRINTER(char4)
```

```
NATLANG(K) PRINTERCOPY(NO) PRINTERCOPY(YES)
```

```
REMOTESYSNET(char8) REMOTESYSTEM(char4) REMOTENAME(char4)
```

```
SECURITYNAME(char8)
```

```
SOLICITED(NO) TASKLIMIT(NO) TERMPRIORITY(0)
```

```
SOLICITED(YES) TASKLIMIT(1-32767) TERMPRIORITY(0-255)
```

```
TERMPRIORITY(255) TRANSACTION(char4)
```

```
TYPETERM(char8) USERID(char8)
```

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 may 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
CREATE TERMINAL

TERMINAL attributes. See "ATTRIBUTES option" on page 26 for general rules for specifying attributes, and the TERMINAL chapter in 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
specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

LOG The resources attributes are logged to the CSDL transient data queue.

NOLOG The resources attributes are not logged to the CSDL transient data queue.

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, "EXEC CICS CREATE RESP2 values," on page 681 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 SYNONRETURN 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(data-value)—ATTRIBUTES(data-value)—ATTRLEN(data-value)—LOGMESSAGE(cvda)

CREATE TRANCLASS attribute values:

<table>
<thead>
<tr>
<th>DESCRIPTION(char58)</th>
<th>MAXACTIVE(0-999)</th>
<th>PURGETHRESH(NO)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PURGETHRESH(1-1000000)</td>
</tr>
</tbody>
</table>

Conditions: ILLLOGIC, 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 “ATTRIBUTES option” on page 26 for general rules for specifying attributes, and the TRANCLASS chapter in 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.
CREATE TRANCLASS

LOGMESSAGE
specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

LOG       The resources attributes are logged to the CSDL transient data queue.

NOLOG     The resources attributes are not logged to the CSDL transient data queue.

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, “EXEC CICS CREATE RESP2 values,” on page 681 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**

\[\text{CREATE TRANSACTION}(\text{data-value})\] — \[\text{ATTRIBUTES}(\text{data-value})\] — \[\text{ATTRLEN}(\text{data-value})\] — \[\text{LOGMESSAGE}(\text{cvda})\]

**CREATE TRANSACTION attribute values:**

- **ACTION**
  - BACKOUT
  - COMMIT
- **ALIAS**
  - char4
- **BREXIT**
  - char8
- **CMDSEC**
  - NO
  - YES
- **CONFDATA**
  - NO
  - YES
- **DESCRIPTION**
  - char58
- **DTIMOUT**
  - NO
  - 1-6800
  - 1-240000
- **DUMP**
  - YES
  - NO
- **DYNAMIC**
  - NO
  - YES
- **INDOUBT**
  - BACKOUT
  - COMMIT
  - WAIT
- **ISOLATE**
  - YES
  - NO
- **LOCALQ**
  - NO
  - YES
- **INDOUBT**
  - BACKOUT
  - COMMIT
  - WAIT
- **ISOLATE**
  - YES
  - NO
- **LOCALQ**
  - NO
  - YES
- **DTIMOUT**
  - NO
  - 1-6800
  - 1-240000
- **PARTITIONSET**
  - char8
  - KEEP
  - OWN
- **PRIORITY**
  - 1
  - 0-255
- **PROFILE**
  - DFHCICST
  - char8
- **PROGRAM**
  - char8
- **REMTESYSTEM**
  - char4
- **REMTENAME**
  - char8
- **RUNAWAY**
  - SYSTEM
  - 0
  - 500-2700000
- **SHUTDOWN**
  - DISABLED
  - ENABLED
- **SPURGE**
  - NO
  - YES
- **STATUS**
  - ENABLED
  - DISABLED
- **STORAGECLEAR**
  - NO
  - YES
- **TASKDATAKEY**
  - USER
  - CICS
  - ANY
- **TASKDATALOC**
  - BELOW
  - ANY
- **TRANCLASS**
  - DFHTCL00
  - char8
- **TPNAME**
  - char64
- **TPRNAME**
  - hex128
- **TWASIZE**
  - 0
  - 0-32767
- **WAIT**
  - YES
  - NO
- **WAITTIME**
  - 0,0,0
  - 0-99,0-23,0-59
- **TPURGE**
  - NO
  - YES
- **TRACE**
  - YES
  - NO
- **TRPROF**
  - DFHCICSS
  - char8
- **XTRANID**
  - hex8

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

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 “ATTRIBUTES option” on page 26 for general rules for specifying attributes, and the TRANSACTION chapter in 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

specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

LOG      The resources attributes are logged to the CSDL transient data queue.

NOLOG    The resources attributes are not logged to the CSDL transient data queue.

TRANSACTION(data-value)

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

Conditions

ILLLOGIC

RESP2 values:
2  The command cannot be executed because an earlier CONNECTION or TERMINAL pool definition has not yet been completed.

INVREQ

RESP2 values:
CREATE TRANSACTION

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, "EXEC CICS CREATE RESP2 values," on page 681 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

CREATE TSMODEL

Define a temporary Storage Table in the local CICS region.

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 may 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 being added. The list of attributes must be coded as a single character string using the syntax shown in TSMODEL attributes. See "ATTRIBUTES option" on page 26 for general rules for
CREATE TSMODEL

specifying attributes, and the TSMODEL chapter in 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**
specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

- **LOG** The resources attributes are logged to the CSDL transient data queue.
- **NOLOG** The resources attributes are not logged to the CSDL transient data queue.

**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, “EXEC CICS CREATE RESP2 values,” on page 681](#) 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.
CREATE TYPETERM

Define a terminal type in the local CICS region.

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

CREATE TYPETERM attribute values (part 1 of 2):

- **ALTPAGE(0,0)**
- **ALTPAGE(0-255,0-255)**
- **ALTSUFFIX(char1)**
- **APLKYBD(NO)**
- **APLTEXT(NO)**
- **ASCII(NO)**
- **ATI(NO)**
- **AUDIBLEALARM(NO)**
- **AUTOCONNECT(NO)**
- **AUTOPAGE(NO)**
- **BACKTRANS(NO)**
- **BACKTRANS(YES)**
- **BUILDCHAIN(NO)**
- **BUILDCHAIN(YES)**
- **CGCSGID(0,0)**
- **CGCSGID(0-65535,0-65535)**
- **COLOR(NO)**
- **COLOR(YES)**
- **COPY(NO)**
- **COPY(YES)**
- **CREATESESS(NO)**
- **CREATESESS(YES)**
- **DEFSCREEN(0,0)**
- **DEFSCREEN(0-255,0-255)**
- **DESCRIPTION(char58)**
- **DEVICE(char8)**
- **DISCREQ(NO)**
- **DISCREQ(YES)**
- **DUALCASEKYBD(NO)**
- **DUALCASEKYBD(YES)**
- **ERRCOLOR(NO)**
- **ERRCOLOR(BLUE)**
- **ERRCOLOR(GREEN)**
- **ERRCOLOR(NEUTRAL)**
- **ERRCOLOR(PINK)**
- **ERRCOLOR(RED)**
- **ERRCOLOR(TURQUOISE)**
- **ERRCOLOR(YELLOW)**
- **ERRINTENSIFY(NO)**
- **ERRINTENSIFY(YES)**
- **ERRHILIGHT(NO)**
- **ERRHILIGHT(BLINK)**
- **ERRHILIGHT(REVERSE)**
- **ERRHILIGHT(UNDERLINE)**
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 "ATTRIBUTES option" on page 26 for general rules for specifying attributes, and the TYPETERM chapter in 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
specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

LOG The resources attributes are logged to the CSDL transient data queue.

NOLOG The resources attributes are not logged to the CSDL transient data queue.

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
CREATE TYPETERM

during either the discard or resource definition phase of the processing.
See Appendix C, “EXEC CICS CREATE RESP2 values,” on page 681 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 in the local CICS region.

CREATE URIMAP

CREATE URIMAP attribute values:

- DESCRIPTION(char58)
- STATUS(ENABLED) STATUS(DISABLED)
- HOST(char116) PATH(char256) SCHEME(HTTP) SCHEME(HTTPS)

SERVER attributes:

- TCPIPSERVICE(char8)
- MEDIATYPE(char40) CHARACTERSET(char40) HOSTCODEPAGE(char16)
- TEMPLATENAME(char48) HFSFILE(char255)
- ANALYZER(YES) ANALYZER(NO)
- CONVERTER(char8) TRANSACTION(char4) PROGRAM(char8)
- USERID(char8) TYPE(NONE) TYPE(Temporary) TYPE(PERMANENT)
- LOCATION(char256)

CLIENT attributes:

- CERTIFICATE(char32) CIPHERS(char48)

PIPELINE attributes:

- PIPELINE(char8) WEBSERVICE(char32) TCPIPSERVICE(char8) TRANSACTION(char4)
- USERID(char8) TYPE(NONE) TYPE(Temporary) TYPE(PERMANENT)
- LOCATION(char256)

Conditions: ILLOGIC, 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 definition. It does not use a resource definition stored in the CSD. If there is already a URIMAP definition with the name you 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 syncpoint 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 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 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 "ATTRIBUTES option" on page 26 for general rules for specifying attributes, and the URIMAP chapter in 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
specifies whether CICS logs the attributes used for the resource being created to the CSDL transient data queue. CVDA values are:

LOG The resources attributes are logged to the CSDL transient data queue.

NOLOG The resources attributes are not logged to the CSDL transient data queue.

URIMAP(data-value)
specifies the 8-character name of the URIMAP 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
CREATE URIMAP

error. See Appendix C, “EXEC CICS CREATE RESP2 values,” on page 681 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 WEBSERVICE

Define a WEBSERVICE in the local CICS region.

CREATE WEBSERVICE

--- CREATE WEBSERVICE(data-value)---ATTRIBUTES(data-value)---ATTRLEN(data-value)---LOGMESSAGE(cvda)---

CREATE WEBSERVICE attribute values:

--- DESCRIPTION(text)---WEBSERVICE(WEBSERVICEName)---WSBIND(hfsfile)---WSDLFILE(hfsfile)---

--- VALIDATION(NO)---VALIDATION(YES)---

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 definition with the attribute specified on the command. It does not use a resource definition stored in the CSD. If there is already a WEBSERVICE 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 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 being added. The list of attributes must be coded as a single character string using the syntax shown in **WEBSERVICE attributes**. See "ATTRIBUTES option" on page 26 for general rules for specifying attributes, and the WEBSERVICE chapter in the CICS Resource Definition Guide for details about specific attributes.
CREATE WEBSERVICE

**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.

**WEBSERVICE**(data-value)

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

**Conditions**

**INVREQ**

RESP2 values:

1. **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, “EXEC CICS CREATE RESP2 values,” on page 681 for information on RESP2 values.

2. **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.

6. **612** Installation of this WEBSERVICE 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.
DISABLE PROGRAM

Terminate or otherwise modify the invocation of a 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

**DISABLE PROGRAM**

<table>
<thead>
<tr>
<th>ENTRYNAME(data-value)</th>
<th>EXIT(data-value)</th>
<th>FORMATEDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>PURGEABLE</td>
<td>SHUTDOWN</td>
<td>SPI</td>
</tr>
<tr>
<td>STOP</td>
<td>TASKSTART</td>
<td></td>
</tr>
</tbody>
</table>

**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 the [CICS Customization Guide](#CICS_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.
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; for programming information, including a list of exit points, see the [CICS Customization Guide](#).

**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 actually 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 whilst 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
DISABLE PROGRAM

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.
DISABLE PROGRAM

Examples

**Example 1**

EXEC CICS DISABLE PROGRAM('EP2') STOP

Example 1 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.

**Example 2**

EXEC CICS DISABLE ENTRYNANE ('ZX') PROGRAM('EP3')
EXIT('XTDREQ')

Example 2 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.

**Example 3**

EXEC CICS DISABLE PROGRAM('EP3') EXITALL

Example 3 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.
DISCARD AUTINSTMODEL

Remove a terminal autoinstall model definition.

**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.

See [“Discarding resource definitions” on page 27](#) for general information about discards.
DISCARD CONNECTION

Remove a CONNECTION definition.

**DISCARD CONNECTION**

<table>
<thead>
<tr>
<th>DISCON1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISCARD CONNECTION(data-value)</td>
</tr>
</tbody>
</table>

**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 VTAM 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.
DISCARD CONNECTION

See “Discarding resource definitions” on page 27 for general information about DISCARD commands.

Options

CONNEXION(*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

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.

**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.

**Note:** 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.

This command is threadsafe.
DISCARD DB2ENTRY

Remove a DB2ENTRY definition.

**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**

```plaintext
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**

```plaintext
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

DISCARD DJAR

Remove the definition of a specified deployed JAR file from the system, together with any associated beans.

DISCARD DJAR

\[ \text{DISCARD DJAR}(\text{data-value}) \]

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 wish 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.

**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.

**NOTFND**

RESP2 values:

- **1** The specified DOCTEMPLATE is not installed on this system.

This command is threadsafe.
DISCARD ENQMODEL

Remove an ENQMODEL resource definition.

DISCARD ENQMODEL

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.

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
DISCARD FILE

Remove a FILE definition.

---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 JOURNALMODEL

Remove a journal model definition.

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**

```plaintext
<table>
<thead>
<tr>
<th>DISCARD</th>
<th>DISCJ1</th>
</tr>
</thead>
</table>
```

**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 DFHJnn, where nn is the two-digit journal number, in the range 01–99. (DFHJ01 is a user journal in CICS Transaction Server for OS/390, not the system log.)
DISCARD JOURNALNAME

Conditions

INVREQ
RESP2 values:
3 The journal specified cannot be discarded.

JIDERR
RESP2 values:
1 The journal cannot 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 PARTNER

Remove a PARTNER definition.

<table>
<thead>
<tr>
<th>DISCARD PARTNER</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISCARD PARTNER(data-value)</td>
</tr>
</tbody>
</table>

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.

**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

<table>
<thead>
<tr>
<th>DISCARD—PROCESSTYPE(data-value)</th>
</tr>
</thead>
</table>

Conditions: INVREQ, NOTAUTH, PROCESSERR

Description

The DISCARD PROCESSTYPE command removes a CICS business transaction services (BTS) PROCESSTYPE definition from the local CICS region.

Notes:

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
  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.

**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.

See **[Discarding resource definitions](#)** on page 27 for general information about discards.
DISCARD REQUESTMODEL

Remove a request model definition.

DISCARD REQUESTMODEL

DICRM1

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 TCPIPSERVICE

Remove a TCPIPSERVICE definition.

DISCARD TCPIPSERVICE

| TCPIPSERVICE(data-value) |

Conditions: INVREQ, NOTAUTH, NOTFND

Description

The DISCARD TCPIPSERVICE command removes a TCPIPSERVICE definition from the local CICS system.

You cannot discard a TCPIPSERVICE 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

TCPIPSERVICE(data-value)

specifies the 8-character name of the TCPIPSERVICE that you want to remove.

Conditions

INVREQ

RESP2 values:

9  The TCPIPSERVICE is still open.
16  The TCPIPSERVICE 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

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 DELETEQ 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 DELETEQ. 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

<table>
<thead>
<tr>
<th>DISTER1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISCARD TERMINAL(data-value)</td>
</tr>
</tbody>
</table>

Conditions: INVREQ, NOTAUTH, TERMDERR

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.
DISCARD TERMINAL

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.

**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 'DFHTCLnn', 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**

- **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.

---

**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.
**DISCARD TSMODEL**

**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

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:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>The URIMAP is not disabled.</td>
</tr>
<tr>
<td>200</td>
<td>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.</td>
</tr>
</tbody>
</table>

NOTAUTH

RESP2 values are:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>The user associated with the issuing task is not authorized to use this command.</td>
</tr>
</tbody>
</table>

NOTFND

RESP2 values are:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>The URIMAP cannot be found.</td>
</tr>
</tbody>
</table>
DISCARD WEBSERVICE

Remove a WEBSERVICE definition.

DISCARD WEBSERVICE

DISCARD WEBSERVICE(data-value)

Conditions: INVREQ, NOTAUTH, NOTFND

Description

Use the DISCARD WEBSERVICE to remove an WEBSERVICE from your CICS region. The WEBSERVICE must be disabled before it can be discarded.

See "Discarding resource definitions" on page 27 for general information about discards.

Options

WEBSERVICE(data-value)

specifies the 8-character name of the WEBSERVICE whose definition is to be discarded.

Conditions

INVREQ

RESP2 values:

22  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:

1  The WEBSERVICE cannot be found
**ENABLE PROGRAM**

Enable a user exit to allow it to be invoked.

---

**Description**

The initial ENABLE PROGRAM command for an exit:
- Defines it as an exit to the executing CICS system and names it
- Sets the initial execution status (whether it is available for execution and the
  points at which it is executed)
- Allocates work areas
- Loads the associated load module if necessary and establishes the entry point
  within it 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 that command on
page "DISABLE PROGRAM" on page 123 for the correspondence between options
on the two commands.

For programming information about exits in CICS, see the *CICS Customization*
Guide; you should also see the general discussion of commands that modify exits
in "Exit-related commands" on page 28.
ENABLE PROGRAM

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. For programming information about exits, and a list of exit
points, see the CICS Customization Guide.

You can name only one exit point on each ENABLE command. If the same exit
is to be invoked from multiple exit points, you need 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.
ENABLE PROGRAM

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 in-doubt protocol. For information about the in-doubt protocol, see the CICS Customization Guide.

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**

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** (task-related user exits only)

specifies that the task-related user exit program is quasi-reentrant, and relies on the serialization provided by CICS when accessing shared resources. The task-related 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 task-related user exit under the QR TCB. If the task-related user exit program uses MVS services, 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 wish. (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 (task-related user exits only)**
specifies that the 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 task-related 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. (If OPENAPI is also specified, and when it is enabled, 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.
### ENABLE PROGRAM

**X'802000'**
The exit is already enabled. ENTRY, LINKEDITMODE, TALENGTH, GALENGTH and GAENTRY 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.

## Examples

### Enabling global user exits

#### Example 1

```cics
EXEC CICS ENABLE PROGRAM('EP') ENTRYNANE('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

```cics
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

```cics
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

```plaintext
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.
Obtain the address and length of a global work area.

**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.

**GASET**(*ptr-ref*)

returns the address of the global work area.

**Note:** If a **GALENGTH** greater than 32767 has been defined (see "**GALENGTH** for ENABLE PROGRAM" on page 159 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.

**Conditions:** INVEXITREQ, NOTAUTH
EXTRACT EXIT

**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 671.

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 only deals with the URIMAP, PIPELINE, and WEBSERVICE resources, to COLLECT STATISTICS for other resources see "COLLECT STATISTICS" on page 38.

EXTRACT STATISTICS performs a function equivalent to COLLECT STATISTICS for the resources URIMAP, PIPELINE, and WEBSERVICE. These could not be provided by extending COLLECT STATISTICS due to a design limitation of that command. The syntax of EXTRACT STATISTICS is different and provides for unlimited future expansion.

```
EXTRACT STATISTICS

 Extract STATISTICS

 RESTYPE (cvda)

 RESID (data-area)

 RESIDLEN (data-value)

 SET (ptr-ref)

 Conditions: INVREQ, IOERR, LENGERR, NOTAUTH, NOTFND

 last time:

 LASTRESET(data-area)

 hours

 minutes

 seconds

 hours:

 LASTRESETHRS(data-area)

 minutes:

 LASTRESETMIN(data-area)

 seconds:

 LASTRESETSEC(data-area)

```


EXTRACT STATISTICS

Description

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

The statistics that CICS gives you 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 should 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. See "Programming interface information" on page 734 for an explanation of 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 not available for the resource type; similarly, where there is no entry in the specific statistics column, you cannot get statistics for an individual resource.

<table>
<thead>
<tr>
<th>Resource type</th>
<th>CVDA</th>
<th>RESIDLEN</th>
<th>Statistic type</th>
<th>Global statistics</th>
<th>Specific statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIPELINE</td>
<td>1124</td>
<td>Char(8)</td>
<td>PIPELINE</td>
<td>—</td>
<td>DFHPIRDS</td>
</tr>
<tr>
<td>URIMAP</td>
<td>1173</td>
<td>Char(8)</td>
<td>URIMAP</td>
<td>DFHWBGDS</td>
<td>DFHWBRDS</td>
</tr>
<tr>
<td>WEBSERVICE</td>
<td>1174</td>
<td>Char(32)</td>
<td>WEBSERVICE</td>
<td>—</td>
<td>DFHPIW DS</td>
</tr>
</tbody>
</table>

Copybooks are provided in ASSEMBLER, COBOL, and PL/I. (There is no copybook for C.) The names of the copybooks are the same in each language. You can find them in the following libraries:

<table>
<thead>
<tr>
<th>Language</th>
<th>Copybook Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSEMBLER</td>
<td>CICSTS31.CICS.SDFHMAC</td>
</tr>
<tr>
<td>COBOL</td>
<td>CICSTS31.CICS.SDFHCOB</td>
</tr>
<tr>
<td>PL/I</td>
<td>CICSTS31.CICS.SDFHPL1</td>
</tr>
</tbody>
</table>

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

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).

RESTYPE(cvda)
requests statistics for a particular resource type depending on the cvda
supplied. Valid CVDA values are:

PIPELINE requests statistics for a PIPELINE; RESID identifies the
  particular PIPELINE.

URIMAP requests statistics for a URIMAP; RESID identifies the particular
  URIMAP.

WEBSERVICE requests statistics for a WEBSERVICE; RESID identifies the
  particular WEBSERVICE.

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.

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 167.

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.

Conditions

INVREQ
  RESP2 values:
  5 An invalid RESTYPE has been specified. Valid types are the cvda
      values listed in Table 2 on page 167.
  6 A mandatory RESID has NOT been specified for the requested
      RESTYPE.
IOERR
  RESP2 values:
  3   The requested statistics area was not functioning. This happens if, for
      instance, statistics control blocks are overwritten.

LENGERR
  RESP2 values:
  7   An invalid RESIDLEN was supplied for the requested RESID.

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).

Examples

e.g.
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
makes use of virtually all the options described in this section. This set of programs
illustrates ways of using the EXTRACT STATISTICS and INQUIRE commands of
CICS Transaction Server for z/OS 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 on installing and operating the
DFH0STAT application. The source code for the application can be found in
CICSTS31.CICS.SDFHSAMP.
INQUIRE AUTINSTMODEL

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 21 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.

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.

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
INQUIRE AUTOINSTALL

**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.

Options

\[\text{BEAN} (\text{data-value})\]
- Specifies the name of the enterprise bean.

\[\text{CORBASERVER} (\text{data-area})\]
- Specifies a 4-character area to receive the name of the associated CorbaServer.

\[\text{DJAR} (\text{data-area})\]
- Specifies a 32-character area to receive the name of the deployed JAR file that contains the bean.

Conditions

\[\text{END}\]
- RESP2 values:
  - 2: There are no more beans to browse.

\[\text{NOTAUTH}\]
- RESP2 values:
  - 100: The user associated with the issuing task is not authorized to use this command.

\[\text{NOTFND}\]
- RESP2 values:
  - 1: The BEAN was not found.

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
INQUIRE BEAN

Examples

EXEC CICS INQUIRE BEAN(*)
EXEC CICS INQUIRE BEAN(*) CORBASERVER(CORB)
EXEC CICS INQUIRE BEAN(*) DJAR(CART)
EXEC CICS INQUIRE BEAN(MyBean) CORBASERVER(CORB)

In these examples, CICS displays information about:
1. All installed beans
2. All beans installed in CorbaServer CORB
3. All beans installed from DJAR CART
4. Installed bean MyBean in CorbaServer CORB.
INQUIRE BRFACILITY

Retrieve information about a virtual terminal (bridge facility) used by the 3270 bridge mechanism.

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 21] 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.

KEEPETIME(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 keepetime specified when the facility was allocated,
INQUIRE BRFACILITY

- or a default value of 5 minutes if no keeptime was specified,

if the keeptime, so determined, is larger than the value of SIT
  BRMAXKEEPETIME it is reduced to BRMAXKEEPETIME.

- If the facility being displayed is a Web-Bridge, the keeptime is initially set to
  be the Webdelay terminal keep time (the second part of the WEBDELAY SIT
  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
  CICSpelix 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:
INQUIRE BRFACILITY

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\(\texttt{data-area}\)
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 is blank if the bridge facility is currently not in use.

USERID\(\texttt{data-area}\)
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 CFDTPOOL

Retrieve information about a coupling facility data table pool.

INQUIRE CFDTPOOL

\[\text{INQUIRE CFDTPOOL}(\text{data-value})\text{ CONNSTATUS}(\text{cvda})\]

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 21 for general information about browsing, including syntax, exception conditions, and examples.

Options

\[\text{CFDTPOOL}(\text{data-value})\]

specifies the 8-character name of the coupling facility data table pool about which you are inquiring.

\[\text{CONNSTATUS}(\text{cvda})\]

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:
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 in the CICS region, and report the presence of any old shared class caches that are awaiting deletion.

AUTOSTARTST cvda returns a CVDA value indicating the status of autostart for the shared class cache. CVDA values are:

**ENABLED**
If the shared class cache has been stopped by an explicit command or has not yet been started on this CICS execution, the shared class cache is started as soon as CICS receives a request to run a Java application in a JVM whose profile requires the use of the shared class cache.

**DISABLED**
If the shared class cache has been stopped by an explicit command or has not yet been started on this CICS execution, an explicit EXEC CICS PERFORM CLASSCACHE INITIALIZE(START) command (or the equivalent CEMT 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 whose profile requires the use of the shared class cache, the request fails.

CACHEFREE data-area returns a doubleword binary value representing the amount of free space in the shared class cache, in bytes.

Conditions: NOTAUTH

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 in the CICS region. There can be only one active shared class cache in a CICS region, therefore no name or identifier is required on this command.

Options

AUTOSTARTST cvda
returns a CVDA value indicating the status of autostart for the shared class cache. CVDA values are:

**ENABLED**
If the shared class cache has been stopped by an explicit command or has not yet been started on this CICS execution, the shared class cache is started as soon as CICS receives a request to run a Java application in a JVM whose profile requires the use of the shared class cache.

**DISABLED**
If the shared class cache has been stopped by an explicit command or has not yet been started on this CICS execution, an explicit EXEC CICS PERFORM CLASSCACHE INITIALIZE(START) command (or the equivalent CEMT 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 whose profile requires the use of the shared class cache, the request fails.

CACHEFREE data-area
returns a doubleword binary value representing the amount of free space in the shared class cache, in bytes.
INQUIRE CLASSCACHE

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 option returns the size that will be used by default when the shared class cache is started. If the status of the shared class cache is STARTING or STARTED, this option returns the size of the current shared class cache. If the status of the shared class cache is RELOADING, this option returns 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 worker JVMs that are dependent on them to be phased out. If the status of the current shared class cache is STOPPED, and worker JVMs are still dependent on it, then 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 worker JVMs that are dependent on an old shared class cache, and are being phased out. If the status of the current shared class cache is STOPPED, then any worker JVMs that are still dependent on it are included in the number of worker JVMs being phased out.

PROFILE(*data-area*)
If the status of the shared class cache is STOPPED, this option returns the eight-character name of the JVM profile that will be used for a master JVM to start the shared class cache. If the status of the shared class cache is STARTED, STARTING or RELOADING, this displays the eight-character name of the JVM profile that was used for the last valid request to start or reload the shared class cache. This name is displayed even if the shared class cache fails to start or unload. The displayed JVM profile is used next time you issue the command to start or reload the shared class cache, unless you specify a different JVM profile using the PROFILE option.

REUSEST(*cvda*)
returns a CVDA value indicating the level of reusability for the master JVM that initializes the shared class cache. All the worker JVMs in a CICS region inherit their level of reusability from the master JVM. CVDA values are:

- **RESET**: The master JVM and worker JVMs are resettable.
- **REUSE**: The master JVM and worker JVMs are continuous.
- **UNKNOWN**: The level of reusability for the master JVM and worker JVMs is not known, because the shared class cache is not started.

STARTTIME(*data-area*)
returns the time when the current shared class cache was started. The time is in ABSTIME format, which is an 8-byte packed decimal field containing the time since 00:00 hours on 1 January 1900 (in milliseconds rounded to the nearest hundredth of a second). 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:
INQUIRE CLASSCACHE

STARTING
The shared class cache is being initialized. If autostart is enabled, the shared class cache is starting either because CICS received a request to run a Java application in a JVM whose profile requires the use of the shared class cache, or because an explicit EXEC CICS PERFORM CLASSCACHE INITIALIZE(START) command (or the equivalent CEMT command) was issued. If autostart is disabled, the shared class cache is starting because an explicit EXEC CICS PERFORM CLASSCACHE INITIALIZE(START) command (or the equivalent CEMT command) was issued. Worker 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 worker JVMs fail.

STARTED
The shared class cache is ready, and it can be used by worker JVMs.

RELOADING
An EXEC CICS PERFORM CLASSCACHE INITIALIZE(RELOAD) command (or the equivalent CEMT command) has been issued, and a new shared class cache is being loaded to replace the existing shared class cache. Worker JVMs, 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 execution, or it has been stopped by an EXEC CICS PERFORM CLASSCACHE TERMINATE command or an EXEC CICS SET JVMPOOL TERMINATE command (or the equivalent CEMT commands). If autostart is disabled, requests to run a Java application in a JVM whose profile requires the use of the shared class cache (that is, requests for worker JVMs) will fail. If autostart is enabled, a new shared class cache will be initialized as soon as CICS receives a request to run a Java application in a JVM whose profile requires the use of the shared class cache.

TOTALJVMS(data-area)
returns a fullword binary value giving the number of worker JVMs in the CICS region that are dependent on a shared class cache. This includes both the worker JVMs that are dependent on the current shared class cache, and any worker JVMs that are dependent on 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 a system 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. A CONNECTION definition is sometimes known as a “system entry”.

There are two main types of system connection:

- Multiregion operation (MRO), which uses the CICS interregion communication program (DFHIRP) to establish a connection between two MRO partners. An MRO connection can exist between two CICS regions on the same MVS image or within the same MVS/ESA sysplex.

- A special form of MRO connection, used by the external CICS interface (EXCI), can exist between a CICS region and a non-CICS client program running in MVS, for example an MVS batch or TSO program. An EXCI connection connects the client program to a CICS region running in the same MVS image or within the same MVS/ESA sysplex.
Intersystem communication (ISC) connections, between CICS and any other system which supports VTAM APPC or LUTYPE6.1 communications. For example, ISC connections can exist between CICS regions running in different MVS/ESA sysplexes or on different operating system platforms, between CICS and any APPC device, and between CICS and IMS.

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 actually owned by another CICS system. The owning system is called the terminal-owning region (TOR). Note that different connections may have different TORs. TOR here refers to the owning system 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 option usually names the CONNECTION definition for the link. (It can name an indirect connection, but that is an unusual setup). In this case, the netname of the TOR is specified in the link CONNECTION definition. The REMOTESYSNET option of the remote CONNECTION definition may or may not specify the netname of the TOR.

If a remote connection is on a system that is not directly linked to the TOR, the REMOTESYSTEM option can name one of two types of connection, as follows:

- 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 Figure 1 on page 186, 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 Figure 1 on page 186, connection CON1 points to connection INDC, which in turn points to INTS. The REMOTESYSNET option of the remote CONNECTION definition may or may not specify the netname of the TOR.

Note, however, that for 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 are true:
  - The system returned in the REMOTESYSTEM field has not been installed.
  - REMOTESYSNET was not specified on the CONNECTION 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.
You can also browse through all of 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 21 for general information about browsing, including syntax, exception conditions, and examples.

Options

**ACCESSMETHOD**(cvda)
returns a CVDA value indicating the type of connection between the local system and the one you are inquiring about. CVDA values are:

**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 may not be on the same MVS image; if it is not, the XCF access method is used when the connection becomes ACQUIRED.

**VTAM**
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 MVS/ESA. XCF is used for MRO links between CICS regions on different MVS images within an MVS 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 may 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. You should use CONNSTATUS in new applications.

**AUTOCONNECT**(cvda) (VTAM 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 VTAM 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:

**ALLCONN**
AUTOCONNECT(ALL) has been specified on the CONNECTION definition. This 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 is to try to bind the LU services manager sessions.
**INQUIRE CONNECTION**

**NONAUTOCONN**
AUTOCONNECT(NO) has been specified for the CONNECTION definition. CICS does not bind LU services manager sessions.

**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:

**ACQUIRED**
The connection is acquired. The criteria for ACQUIRED for VTAM links are:
- The partner LU has been contacted.
- The initial CHANGE-NUMBER-OF-SESSIONS (CNOS) exchange has been done.

The criteria for ACQUIRED for MRO links are:
- 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 there are currently no bound sessions 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 may 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 may also be because:
- The remote CICS region has not yet logged on to DFHIRP.
- The remote CICS region has closed interregion communication.
INQUIRE CONNECTION

In the case of an APPC ISC connection, the RELEASED status may also be because:

- The remote CICS region has not yet opened its VTAM 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:

- **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:

- **COMPLETE**
  The quiesce protocol completed successfully when the connection was released. This reverts to NOTATTEMPTED if the connection is reacquired.

- **FAILED**
  The protocol failed. This can occur 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.

- **NOTATTEMPTED**
  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. This could be, for example, because the partner is a back-level CICS region that does not support the connection quiesce protocol.

```
EXITTRACING(cvda) (VTAM only)
```

returns a CVDA value indicating whether the terminal exit program is tracing the sessions associated with this connection. CVDA values are:

- **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
INQUIRE CONNECTION

resource if the two NETNAMEs sent with a BIND are different. This information may also be returned for a partner which is not a generic resource but which uses XRF.

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 may also be returned for a partner that is not a generic resource but which uses XRF.

NETNAME(data-area)
returns the 8-character name by which the remote system is known to the network (from the NETNAME value specified in the CONNECTION definition).

For an ISC connection, the NETNAME corresponds to the VTAM APPLID of the remote system.

For a CICS-to-CICS MRO connection, the NETNAME is the name the remote system uses to log on to DFHIRP (from the APPLID option in its system initialization table (SIT)).

For a SPECIFIC EXCI connection, NETNAME is the name of the client program which 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 SIT 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 VTAM 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 VTAM. 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-VTAM, NQNAME is blank. If the resource is a VTAM 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:

NOTAPPLIC
This is not an APPC parallel-session nor a CICS-to-CICS MRO connection.

NOTPENDING
There has been no mismatch of lognames with the partner.
INQUIRE CONNECTION

**Note:** MRO connections to pre-CICS Transaction Server for z/OS systems do not use lognames. Therefore, for these connections, PENDSTATUS always returns NOTPENDING.

**PENDING**

There is resynchronization work 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, 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 is an APPC connection, no new syncpoint work (that is, work involving synclevel 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) (VTAM and EXCI only)

returns a CVDA value identifying the protocol in use if this is a VTAM or EXCI connection. CVDA values are:

- **APPC** The connection uses the VTAM 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 VTAM 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 there is resynchronization work outstanding for the connection. The connection may never have been connected, have been quiesced and all resynchronization work completed, or disrupted without quiesce—in which case resynchronization may be necessary. CVDA values are:

- **NORECOVDATA** Neither side has recovery information outstanding.
INQUIRE CONNECTION

NOTAPPLIC
This is not an APPC parallel-session nor a CICS-to-CICS MRO connection, and does not support two-phase commit protocols.

NRS CICS does not have recovery outstanding for the connection, but the partner may have.

RECOVDATA
There are in-doubt units of work associated with the connection, or there are outstanding resyncs awaiting FORGET on the connection. Resynchronization takes place when the connection next becomes active, or when the UOW is unshunted.

If there is recovery outstanding, then on completion of exchange lognames, either resynchronization takes place or, in the case of a cold exchange, the PENDING condition is created.

REMTENAME(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.

REMTESYSNET(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.

REMTESYSTEM(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:

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:

NOTAPPLIC
The XLN process is not applicable. This can be because the link:
• Is released
INQUIRE CONNECTION

- 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) (VTAM only)
returns a CVDA value indicating whether the VTAM control component of CICS is tracing activity on the sessions associated with this connection. CVDA values are:

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.
**INQUIRE CORBASERVER**

**Retrieve information about a particular CorbaServer.**

---

### Options

- **ASSERTED**(data-area)
  - returns the 8-character name of a TCPIPSERVICE resource that defines the characteristics of the port which 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:

---

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.

**Description**

The **INQUIRE CORBASERVER** command allows you 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 "[Browsing resource definitions]" on page 21 for general information about browsing, including syntax, exception conditions, and examples.

**Conditions:** END, NOTAUTH, NOTFND
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 within 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 within this certificate is used to provide inputs to the Distinguished Name URM (DFHEJDNX). See the CICS Intercommunication Guide.

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 which is used for inbound IIOP with SSL client certificate authentication.

CORBASERVER(data-value)
specifies the 4-character name of a CorbaServer.

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:

DISABLED
The CorbaServer is currently not processing any requests and is unable to accept new requests. It may have failed to initialize properly or been explicitly disabled.

DISABLING
The CorbaServer is quiescing before entering disabled state. It is not accepting new requests but is allowing currently-executing work to 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 currently-executing 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.
**INQUIRE CORBASERVER**

HOST(data-area)
returns a 255-character area containing the TCP/IP host name, or a string containing the dotted decimal TCP/IP address, which is included in Interoperable Object References (IORs) exported from the CorbaServer.

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:

**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 through 143999 (99 days, 23 hours, 59 minutes) of inactivity after which a session bean may 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 TCPIPSERVICE resource that defines the characteristics of the port which is used for inbound IIOP with SSL but no client authentication.

UNAUTH(data-area)
returns the 8-character name of a TCPIPSERVICE resource that defines the characteristics of the port which 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.

Description

The INQUIRE DB2CONN command allows you to inquire about attributes of the currently installed DB2CONN which defines the connection to DB2.

Note that because there can be only one DB2CONN installed at a time, the name of the DB2CONN is not required on input.

Conditions: NOTAUTH, NOTFND

This command is threadsafe.
Options

ACCOUNTREC
returns the minimum amount of DB2 accounting required for transactions using pool threads. The specified minimum may be exceeded as described in the following options. CVDA values are:

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 the thread is released at syncpoint) may use a different thread for each of its UOWs. The result may be the production of an accounting record for each UOW. For example, an accounting record is produced if a thread terminates 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 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 will use a different thread for each UOW (assuming the thread is released at syncpoint). In this case an accounting record may be produced for each UOW. For example, an accounting record is produced if a thread terminates 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 terminated. Additionally, authorization changes cause accounting records to be produced.

AUTHID
returns an id to be used for security checking when using pool threads. If an AUTHID is returned, AUTHTYPE is not applicable.

AUTHTYPE
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:

GROUP
The 8-character USERID and the connected group name are used as the authorization ID. The following table shows how these two values are interpreted by DB2.

<table>
<thead>
<tr>
<th>IDs passed to DB2</th>
<th>How DB2 interprets values</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS sign-on user ID (USERID)</td>
<td>Represents the primary DB2 authorization ID.</td>
</tr>
</tbody>
</table>
**INQUIRE DB2CONN**

<table>
<thead>
<tr>
<th>IDs passed to DB2</th>
<th>How DB2 interprets values</th>
</tr>
</thead>
<tbody>
<tr>
<td>RACF-connected group name</td>
<td>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.</td>
</tr>
</tbody>
</table>

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, AUTHTYPE(TERM) should not be used.

**TX**
The transaction identification (four characters padded to eight) is used as the authorization ID.

**OPID**
The user operator identification associated with the userid, 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 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 AUTHTYPE(USERID) and AUTHTYPE(GROUP).

**COMAUTHID**
returns an ID to be used for security checking when using command threads. If COMAUTHType is returned, COMAUTHId is not applicable.

**COMAUTHTYPE**
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:

**CGROUP**
The 8-character USERID and the connected group name are used as the authorization ID. The following table shows how these two values are interpreted by DB2.

<table>
<thead>
<tr>
<th>IDs passed to DB2</th>
<th>How DB2 interprets values</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS sign-on user ID (USERID)</td>
<td>Represents the primary DB2 authorization ID.</td>
</tr>
</tbody>
</table>
In DB2, the following IDs passed to DB2 are interpreted by DB2 according to the following rules:

- **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.

- **CSIGN**: The SIGNID parameter of the DB2CONN 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, COMAUTHTYPE(CTERM) should not be used.

- **CTX**: The transaction identification (four characters padded to eight) is used as the authorization ID.

- **COPID**: 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).

- **CUSERID**: 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 COMAUTHTYPE(CUSERID), 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 COMAUTHTYPE(CUSERID) and COMAUTHTYPE(CGROUP).

### DB2CONN Inquire Commands

- **COMTHREADS**: Returns the current number of active command threads.

- **COMTHREADLIM**: Returns the current maximum number of command threads the CICS DB2 attachment allows active before requests overflow to the pool.

- **CONNECTERROR**: Returns how the fact that CICS is not connected to DB2 because the adapter is in 'standby mode' is reported back to an application that has issued a SQL request. CVDA values are:

  - **ABEND**: The application is abended with abend AEY9.
  - **SQLCODE**: The application receives a -923 SQLCODE.

- **CONNECTST**: Returns the status of the CICS DB2 connection. CVDA values are:

  - **CONNECTED**: CICS is connected to DB2.
  - **NOTCONNECTED**: CICS is not connected to DB2.
INQUIRE DB2CONN

CONNECTING
CICS is currently attempting to connect to DB2.

DISCONNING
CICS is currently disconnecting from DB2.

DB2GROUPID
If you are using group attach, this returns the name of a data sharing group 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 within
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
If you are not using group attach, this 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 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 of that subsystem with the status CONNECTING.

DB2RELEASE
returns a four-character value indicating the version and release level of the
DB2 subsystem CICS is connected to. When CICS is not connected to DB2,
blanks are returned.

DROLLBACK
returns a value showing whether the CICS DB2 attachment is to initiate a
SYNCPOINT ROLLBACK in the event of a transaction being selected as victim
of a deadlock resolution. CVDA values are:

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.

MSGQUEUE1
returns the name of the first transient data destination to which unsolicited
messages from the CICS DB2 attachment are sent.

MSGQUEUE2
returns the name of the second transient data destination to which unsolicited
messages from the CICS DB2 attachment are sent.

MSGQUEUE3
returns the name of the third transient data destination to which unsolicited
messages from the CICS DB2 attachment are sent.
INQUIRE DB2CONN

NONTERMREL
returns a value showing whether non-terminal transactions are to release
threads for reuse at intermediate syncpoints. CVDA values are:

**RELEASE**  non-terminal transactions release threads for reuse at
intermediate syncpoints.

**NORELEASE**  non-terminal transactions do not release threads for reuse at
intermediate syncpoints.

PLAN
returns the name of the plan used for the pool. If a plan name is returned,
PLANEXITNAME is blank.

PLANEXITNAME
returns the name of the dynamic plan exit used for pool threads. If a
PLANEXITNAME is returned, PLAN is blank.

PRIORITY
returns the priority of the pool thread TCBs relative to the CICS main TCB (QR
TCB). If CICS is connected to DB2 Version 6 or later, the thread TCBs are
CICS open L8 TCBs. If CICS is connected to DB2 Version 5 or earlier, the
thread TCBs are private TCBs created by the CICS-DB2 Attachment Facility.
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
returns in minutes the length of the protected thread purge cycle. 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. Hence if the purge cycle is set to 30 seconds after it is released,
a protected thread is purged 30 - 60 seconds after it is released. An
unprotected thread is terminated when it is released (at syncpoint or end of
task) if there are no other transactions waiting for a thread on that DB2ENTRY.

PURGECYCLES
returns in seconds the length of the protected thread purge cycle. The range for
Purgecycles is 30-59.

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. Hence if the purge cycle is set to 30 seconds after it is released,
a protected thread is purged 30 - 60 seconds after it is released. An
unprotected thread is terminated when it is released (at syncpoint or end of
task) if there are no other transactions waiting for a thread on that DB2ENTRY.

RESYNCHMEMBER
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
will occur when CICS has resynchronized with its remote coordinator.) CVDA
values are:

**RESYNC**  CICS connects to the same DB2 data sharing group member.
INQUIRE DB2CONN

**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.

If you perform an INQUIRE DB2CONN RESYNCMEMBER command and are not using group attach, a Not Applicable value is returned.

**SIGNID**  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**  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:

- **NOCONNECT**  The CICS DB2 attachment terminates.
- **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 subsequently fails the CICS DB2 attachment reverts to standby mode again and subsequently reconnects to DB2 when it comes up again.

**STATSQUEUE**  returns the transient data destination for CICS DB2 attachment statistics produced when the CICS DB2 attachment is shutdown.

**TCBS**  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 high-water 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.

**TCBLIMIT**  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
INQUIRE DB2CONN

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.

THREADERROR
returns the processing that is to occur following a create thread error. CVDA values are:

ABEND
For a second or subsequent SQL error the transaction is abended with abend code AD2S, AD2T, or AD2U, dependent on the type of error that occurred. The transaction must be terminated 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. 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. SYNCPOINT without the ROLLBACK option results in an ASP3 or ASP7 abend.

THREADLIMIT
returns the current maximum number of pool threads the CICS DB2 attachment allows active before requests are made to wait or are rejected (see THREADWait).

THREADS
returns the current number of active pool threads.

THREADWAIT
returns whether or not transactions should wait for a pool thread or be abended should the number of active pool threads reach the threadlimit number. CVDA values are:

TWAIT
If all threads are busy, a transaction waits until one becomes available.

NOTWAIT
If all threads are busy, a transaction is terminated 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

ACCOUNTREC(cvda)

AUTHID(data-area)

AUTHTYPE(cvda)

DISABLEDACT(cvda)

DROLLBACK(cvda)

ENABLESTATUS(cvda)

PLAN (data-area)

PLANEXITNAME(data-area)

PRIORITY(cvda)

PROTECTNUM(data-area)

PTHREADS(data-area)

THREADLIMIT(data-area)

THREADS (data-area)

THREADWAIT(cvda)

Conditions: END, ILLOGIC, 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. For RCTs migrated to the CSD, it is identified by the name of the first transaction on the DSNCRCT TYPE=ENTRY statement unless the RDONAME parameter has been specified.

Browsing

You can also browse through all of the DB2ENTRYs installed in a CICS region by using the browse options (START, NEXT, and END) on INQUIRE DB2ENTRY commands. See “Browsing resource definitions” on page 21 for general information about browsing, including syntax, exception conditions, and examples.

Options

ACCOUNTREC

returns the minimum amount of DB2 accounting required for transactions using this DB2ENTRY. The specified minimum may be exceeded, as described in the following options. CVDA values are:

UOW

The CICS DB2 attachment facility causes an accounting 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 may use a different thread for each UOW (assuming the thread is released at syncpoint). The result may be the production of an accounting record for each UOW. For example, an accounting record is produced for each UOW.
produced if a thread terminates 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 DB2ENTRYs that are used by more than one transaction ID. As threads are typically released at syncpoint, a transaction containing multiple UOWs may use a different thread for each UOW. The result may be that an accounting record is produced for each UOW. For example, an accounting record is produced if a thread terminates 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 terminated. 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:

- **GROUP**
The 8-character userid and the connected group name are used as the authorization ID. The following table shows how these two values are interpreted by DB2.

<table>
<thead>
<tr>
<th>IDs passed to DB2</th>
<th>How DB2 interprets values</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS sign-on user ID (USERID)</td>
<td>Represents the primary DB2 authorization ID.</td>
</tr>
<tr>
<td>RACF-connected group name</td>
<td>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.</td>
</tr>
</tbody>
</table>

If no RACF group ID is available for this USERID, then 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, AUTHTYPE(TERM) should not be used.
INQUIRE DB2ENTRY

TX  The transaction identification (four characters padded to eight)
is used 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 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 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 AUTHTYPE(USERID) and
AUTHTYPE(GROUP).

DISABLEDACT
returns what CICS is to do with new transactions accessing the DB2ENTRY
when it has been disabled or disabling. If DISABLEDACT is not specified, and
DB2ENTRY is disabled, new requests are routed to the pool by default. 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 MSGQUEUEn 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.

DROLLBACK
returns whether or not the CICS DB2 attachment should initiate a SYNCPOINT
rollback in the event of a transaction being selected as victim of a deadlock
resolution. CVDA values are:

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 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:

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 using the DB2ENTRY are allowed to complete
unless the DB2ENTRY is being disabled with the FORCE
option.
INQUIRE DB2ENTRY

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). If CICS is connected to DB2 Version 6 or later, the thread TCBs are CICS open L8 TCBs. If CICS is connected to DB2 Version 5 or earlier, the thread TCBs are private TCBs created by the CICS-DB2 Attachment Facility. 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
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 2 purge cycles, the thread is terminated.

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 THREADWait).

THREADWAIT
returns whether or not transactions should wait for a DB2ENTRY thread be abended, or overflow to the pool if 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, 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 terminated with an abend code AD3T.

Conditions

NOTFND
RESP2 values:
1  The DB2ENTRY cannot be found.

NOTAUTH
RESP2 values:
100  Command authorization failure
INQUIRE DB2ENTRY

101  Resource authorization failure
INQUIRE DB2TRAN

INQUIRE DB2TRAN

Returns attributes of a particular DB2TRAN which associates a transaction or group of transactions with a DB2ENTRY.

Description

The DB2TRAN is identified by the name it was defined with in CEDA. For RCTs migrated to the CSD, the name is the same as the transaction for which the DB2TRAN is being created.

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 DB2TRANs installed in your system by using the browse options (START, NEXT, and END) on INQUIRE DB2TRAN commands. See “Browsing resource definitions” on page 21 for general information about browsing, including syntax, exception conditions, and examples.

Options

DB2ENTRY

returns the name of the DB2ENTRY to which this DB2TRAN refers; that is, the DB2ENTRY with which this additional transaction is associated.

PLAN

returns the name of the plan retrieved from the associated DB2ENTRY if it exists. If there is no associated DB2ENTRY, or the DB2ENTRY is disabled with DISABLEDACT(POOL), then 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 the DB2ENTRY is disabled with DISABLEDACT(POOL), then 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

**NOTFND**
RESP2 values:
1 The DB2TRAN cannot be found.

**NOTAUTH**
RESP2 values:
100 Command authorization failure
101 Resource authorization failure
INQUIRE DELETSHIPPED

INQUIRE DELETSHIPPED
Retrieve information about system settings that control the CICS timeout delete mechanism.

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 DELETSHIPPED 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 0hhmmss+. 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.

IDLEMINS(*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 0hhmmss+, 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.

INTERVALMINS(*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.

Description

The INQUIRE DISPATCHER command returns information about the CICS system under which the task issuing the command is executing.

Many of the options in this command correspond to options in the system initialization table (SIT) and take their initial values from the SIT. Some of these can be changed by a subsequent SET DISPATCHER command, or by using its CEMT equivalent. 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.

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
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 only use L8 mode TCBs, for example, the CICS DB2 adaptor when connecting to the DB2 Version 6 or later). 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.

**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 actually 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 actually allocated, see the ACTOPENTCBS option.

The difference between MAXOPENTCBS and ACTOPENTCBS represents the number of such TCBs that are free.

**MAXSSLTCBS**(data-value)

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.

**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 actually 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 executing a transaction whose profile does not specify runaway task time (see the INQUIRE TRANSACTION option "RUNAWAY" on page 395).

**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 option in the system initialization table 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

INQUIRE DJAR
Retrieve information about a DJAR definition.

INQUIRE DJAR

INQUIRE --DJAR(data-value)

-CORBASERVER(data-area)
-HFSFILE(data-area)
-LASTMODTIME(data-area)
-STATE(cvda)

Conditions: NOTAUTH, NOTFND

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" 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 21 for general information about browsing, including syntax, exception conditions, and examples.

Options

CORBASERVER(data-area)
specifies a 4-character area to receive the name of the associated CorbaServer.

DJAR(data-value)
specifies a 1–32-character deployed JAR file name.

HFSFILE(data-area)
specifies a 255-character area to receive the fully-qualified HFS file name.
   If this DJAR definition was created by performing the SCAN function on a CORBASERVER definition, this fully-qualified HFS file name is the one that appears in the DJARDIR parameter of that CORBASERVER definition.

LASTMODTIME(data-area)
specifies an 8-byte packed-decimal field to receive the time, in milliseconds since 00:00 on January 1st 1900, that the deployed JAR file on HFS was last updated. This is a readonly value that CICS updates 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.
INQUIRE DJAR

For dynamically-installed DJARs (those installed by the CICS scanning mechanism), the value of LASTMODTIME is the timestamp of the HFS file pointed to by the DJAR definition, 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 timestamp of the HFS file pointed to by the DJAR definition, 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:

- **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 out whether a DOCTEMPLATE is installed.

**Description**

The INQUIRE DOCTEMPLATE command allows you to determine whether a particular DOCTEMPLATE is installed (defined in the current execution of your CICS system).

**Browsing**

You can also browse through all of the DOCTEMPLATE installed in your system by using the browse options (START, NEXT, and END) on INQUIRE DOCTEMPLATE commands. See "Browsing resource definitions" on page 21 for general information about browsing, including syntax, exception conditions, and examples.

**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:

**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.
INQUIRE DOCTEMPLATE

DDNAME(data-value)
    returns the DD name of the PDS containing the template. The DDName applies
    only to a template of type PDS.

DOCTEMPLATE(data-value)
    specifies the 8-character identifier of the DOCTEMPLATE about which you are
    inquiring.

DSNAME(data-value)
    returns the data set name of the PDS containing the template. It applies only to
    a template of type PDS.

EXITPGM(data-value)
    returns the exit program to be invoked when a request is made for this
    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-value)
    returns the 8-character name of the CICS file definition for the data set
    containing the template.

HFSFILE(data-value)
    returns the fully-qualified name of the z/OS UNIX System Services HFS file
    where the template resides. This can be up to 255 characters in length.

MEMBER(data-value)
    returns the name of the member in the PDS containing the template. MEMBER
    applies only to a template of type PDS.

PROGRAM(data-value)
    returns the program in which the template data is stored. CICS loads the
    program and takes all data after the entrypoint to be the template.

TDQUEUE(data-value)
    returns the name of the TD queue on which the template is stored.

TEMPLATENAMEx(data-value)
    returns the extended template-name by which the doctemplate is to be known
    outside the resource definition function.

TEMPLATETYPE(cvda)
    returns a CVDA value identifying the type of the source of this template. CVDA
    values are:

    EXIT     an exit program.
    FILE     a CICS file name for a data set.
    HFSFILE  a z/OS UNIX System Services HFS file.
    PDSMEMBER a name of the member in the PDS described in DDNAME.
    PROGRAM  a name of a program.
    TDQ      a name of a TD queue.
    TSQ      a name of a TS queue.

TSQUEUE(data-value)
    returns the name of the TS queue on which the template is stored.
INQUIRE DOCTEMPLATE

**TYPE**(data-value)

returns a CVDA value identifying the format of the template contents. CVDA values are:

- **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.

**NOTFND**

RESP2 values:

1  The DOCTEMPLATE specified cannot be found.
INQUIRE DSNAME

Retrieve information about an external data set.

INQUIRE DSNAME

\texttt{INQUIRE DSNAME(data-value)}

- \texttt{ACCESSMETHOD(cvda)}
- \texttt{AVAILABILITY(cvda)}
- \texttt{BACKUPTYPE(cvda)}
- \texttt{BASEDSNAME(data-area)}
- \texttt{FILECOUNT(data-area)}
- \texttt{RECOVSTATUS(cvda)}
- \texttt{FSNRECOVLOG(data-area)}
- \texttt{FSDRECOVLSN(data-area)}
- \texttt{OBJECT(cvda)}
- \texttt{LOSTLOCKS(cvda)}
- \texttt{QUIESCESTATE(cvda)}
- \texttt{RECOVSTATUS(cvda)}
- \texttt{RETOLOCKS(cvda)}
- \texttt{VALIDITY(cvda)}

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 option 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.

\textbf{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 21 for general information about browsing, including syntax, exception conditions, and examples.
INQUIRE DSNAME

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 actually 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
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.
INQUIRE DSNAME

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 DFSMSShsm™ or DFSMSdss, 2 can take a backup copy.

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 DSNAME 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 DSNAME 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), or
- 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.

---

2. Hierarchical storage manager (DFSMShsm) and data set services (DFSMdss) are components of Data Facility Storage Management Subsystem (DFSMS/MVS).
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 UOWs 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.
INQUIRE DSNAME

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:

FWDRECOVABLE
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.

NOTRECOVABLE
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 NOTRECOVABLE 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:
INQUIRE DSNAME

- Has no deferred recovery work for the base data set, and therefore no retained locks, or
- Has recovery work currently in progress.

Note that retained locks may 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 Recovery and Restart Guide](https://www.ibm.com/) 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.

**RETAI**ned

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.
INQUIRE DSNAME

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.

For more information about the use of CVDA, see "CICS-value data areas (CVDA)" 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.
INQUIRE DUMPDS

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 417 for a full description.

Browse only function

The INQUIRE ENQ command, (and the INQUIRE UOWENQ command), can be used only in browse mode.

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.

**Options**

**ENQMODEL**(data-value)

specifies the 8-character identifier of an enqueue model.

**ENQSCOPE**(data-area)

returns the 4-character name which 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.

**STATUS**(cvda)

returns a CVDA value describing the current state of the ENQMODEL. CVDA values are:

- **ENABLED**

  matching enqueue requests are being processed in the normal way.

Conditions: END, ILLOGIC, NOTAUTH, NOTFND

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
INQUIRE ENQMODEL

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. There are INSTALL CREATE or
DISCARD requests 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 EXCI

INQUIRE EXCI
Retrieve information about jobs using the external CICS interface.

**Browse only function**
The INQUIRE EXCI command can be used only in browse mode.

<table>
<thead>
<tr>
<th>INQUIRE EXCI(data-value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TASK(data-value)</td>
</tr>
<tr>
<td>URID(data-value)</td>
</tr>
</tbody>
</table>

**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.
INQUIRE EXCI

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.
Retrieve information about a user exit.

**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:

```
Browse EXITPROGRAM
INQUIRE EXITPROGRAM EXIT(data-value) START
```
INQUIRE EXITPROGRAM

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**(task-related user exits only)

- returns a CVDA indicating which APIs the task-related 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 task-related 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 task-related user exit program, as specified by the latest ENABLE command for this program.

CVDA values are:

- **QUASIRENT**
  - The task-related user 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, this task-related user exit program must switch to a privately-managed TCB.

- **THREADSAFE**
  - The program is defined as threadsafe, and is capable of running under an open TCB. If the APIST option returns OPENAPI, it will always be invoked under an open TCB. If the APIST option returns BASEAPI, it is invoked under whichever TCB is in use by its user task when the program is given control, which 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 EXITPROGRAM

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 CONNECTSTAT 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 ENTRNAME 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
INQUIRE EXITPROGRAM

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 on page 159] 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.
INQUIRE EXITPROGRAM

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.

**TASKSTART(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)" on page 8.

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 returns information about the associated object as well.

- For VSAM, the object may 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 INQUIRE FILE 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:

- Whether the file is open or closed and, if it is closed, whether it ever 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, and so defaults or nulls are returned for some options.
  For further information about null values, see "Null values" on page 14.

- If a file is empty (in VSAM load mode), then, after the first write or massinsert 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 can be a temporary state 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. Many of these parameters can be specified even when the file refers to a different object from that to which the parameters apply. This is intended to make it 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 also browse through all of the files installed in your system by using the browse options (START, NEXT, and END) on INQUIRE FILE commands. See "Browsing resource definitions" on page 21 for general information about browsing, including syntax, exception conditions, and examples.

**Options**

**ACCESSMETHOD** *(cvda)*

returns a CVDA value identifying the access method for this file. CVDA values are:

- **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.
INQUIRE FILE

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:

ADDBABLE
New records can be added to the file.

NOTADDBABLE
New records cannot be added to the 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 DSNM option.

The 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 may not be open at the time the command is issued. This is 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 DSNM. BASEDSNAME is blank for a coupling facility data table that is not associated with a source data set.

Note: The translator still accepts BASENAME for this option, but you should use 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:

BLOCKED
The records on the file are blocked, or this is a VSAM file.

UNBLOCKED
The records on the file are unblocked.

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:

BROWSABLE
You can browse the file.

NOTBROWSABLE
You cannot browse the file.
INQUIRE 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.

**DELETE(cvda) (VSAM only)**
returns a CVDA value identifying whether you can delete records from the file. CVDA values are:

- **DELETABLE**
  You can delete records from the file.

- **NOTDELETEABLE**
  You cannot delete records from the 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:

- **OLD**
  Disposition is OLD.

- **SHARE**
  Disposition is SHARE.

**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, and blanks for a coupling facility data table that is not loaded from a data set.

**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:

- **EMPTYREQ**
  The data set should be made empty.

- **NOEMPTYREQ**
  The data set should not be made empty.

**ENABLESTATUS(cvda)**
returns a CVDA value identifying whether application programs can access the file. CVDA values are:

- **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 executing
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.

**UNENABLING**

A request to close the file has been received, but tasks are executing that had previously accessed the file. These tasks are allowed to complete their use of the file, but new tasks are not allowed access.

**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:

**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.

**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 can be different from the value on the file definition because, for example, the file definition may be overridden by a value from the ICF catalog.

CVDA values are:

**FWDRECOVABLE**

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 DSNNAME.

**NOTFWDRCVBLE**

The file is not forward-recoverable. CICS returns NOTFWDRCVBLE for a coupling facility data table and a user-maintained data table.

**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: There is no autojournaling of requests made to these tables.

**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.

**Notes:**

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:

- **LOAD** The coupling facility data table is, or is to be, 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**)
returns a fullword binary field indicating the number of the VSAM LSR pool associated with this file, in the range 1–8. If the file does not share buffers, this 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 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 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 upon. 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:

- **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 though 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.

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:

- **CLOSED** The file is closed.
CLOSING
The file is in the process of being closed. Closing a file may require
dynamic deallocation of data sets and deletion of shared resources, in
which case close processing may last a significant length of time.

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.

OPENING
The file is in the process of being opened.

READ(cvda)
returns a CVDA value identifying whether you can read records from the file.
CVDA values are:

NOTREADABLE
You cannot read records from the file.

READABLE
You can read records from the 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:

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.

Note: 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.
INQUIRE FILE

RECORDFORMAT(cvda)
returns a CVDA value identifying the format of the records on the file. CVDA values are:

**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.

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 can be different from the value on the file definition because, for example, the file definition may be overridden by a value from the ICF catalog (but any value from the ICF catalog is ignored for a user-maintained or CICS-maintained data table).

CVDA values are:

**NOTRECOVABLE**
  The file is not recoverable.

**RECOVERABLE**
  The file is recoverable.

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:

**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 (MBBCCHHR) 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. CVDA value can be:

  **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:

  **NOTAPPLIC**
  The file is not eligible to be accessed in RLS mode because:
  - It is a remote file, or
  - 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:

  **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.

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, or 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.
INQUIRE FILE

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:

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:

NOTUPDATABLE  You cannot update records.
UPDATABLE  You can update records.

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:

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.

It is possible to 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 the CICS Resource Definition Guide.

Conditions

END

RESP2 values:
2  There are no more resource definitions of this type.

FILENAMEFOUND

RESP2 values:
1  The file cannot be found.
INQUIRE FILE

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.

Description

The INQUIRE HOST command allows you to retrieve information about a particular virtual host in the local CICS region. Virtual hosting in the CICS Internet Guide explains what virtual hosts are. Virtual hosts are based on the URIMAP resource definition object. CICS automatically creates virtual hosts for you, by grouping together into a single data structure all the URIMAP definitions in a CICS region that specify the same host name and the same TCPIPSERVICE. URIMAP definitions that specify no TCPIPSERVICE, and therefore apply to all of them, are added to all the data structures that specify a matching host name, so these URIMAP definitions might be part of more than one data structure.

Browsing

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\textit{(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.

\textbf{Conditions}

\textbf{END}

RESP2 values are:

\begin{itemize}
  \item 2 There are no more virtual hosts.
\end{itemize}

\textbf{ILLOGIC}

RESP2 values are:

\begin{itemize}
  \item 1 You have issued a \texttt{START} command when a browse of this resource type is already in progress, or you have issued a \texttt{NEXT} or an \texttt{END} command when a browse of this resource type is not in progress.
\end{itemize}

\textbf{INVREQ}

RESP2 values are:

\begin{itemize}
  \item 10 The specified host name contains disallowed characters, or is blank.
\end{itemize}

\textbf{NOTAUTH}

RESP2 values are:

\begin{itemize}
  \item 100 The user associated with the issuing task is not authorized to use this command.
\end{itemize}

\textbf{NOTFND}

RESP2 values are:

\begin{itemize}
  \item 5 The virtual host cannot be found.
\end{itemize}
INQUIRE IRC

INQUIRE IRC
Show the IRC status.

INQUIRE IRC

OPENSTATUS(cvda)

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.

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.

Description

The INQUIRE JOURNALMODEL command returns information about a particular installed journal model and enables you to 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 21 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*
INQUIRE JOURNALMODEL

USERJNL1
USERJNL2
USERJNL*

Options

**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 3 of the 4 symbols symbols &USERID, &APPLID, &JNAME and &SYSID.

The name, LSN or template, can be up to 26 characters in length. Names less than 26 character are padded with trailing blanks (X'40').

**TYPE**(cvda)
indicates the log stream type. The CVDA values are:

- **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.

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 21 for general information about browsing, including syntax, exception conditions, and examples.

Options

\[
\text{JOURNALNAME(}data\text{-value}\text{)}
\]

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 DFHJnn, where \( nn \) is the journal number.

To inquire on the system log, specify DFHLOG.

\[
\text{STATUS(}cvda\text{)}
\]

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.

Conditions: END, ILLOGIC, JIDERR, NOTAUTH
**INQUIRE JOURNALNAME**

**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 Transaction Server for OS/390 for retrieving information about the system log and user journals.

Description

For CICS Transaction Server for OS/390, this 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 Transaction Server 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 JVMs in a CICS region and report their status.

**Description**

The INQUIRE JVM command allows you to inquire on the status of a particular JVM, by specifying its JVM token. You can also use the browse function to browse through all of the JVMs in a CICS region, and to identify a JVM about which you want to inquire.

The INQUIRE JVM command does not find the master JVM that initializes the shared class cache. It only finds JVMs that are, or could be, allocated to tasks.

**Browsing**

You can browse through all of the 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 tokens. The browse guarantees that data for each JVM that exists prior to 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 21](#) for general information about browsing, including syntax, exception conditions, and examples.

**Options**

- `JVM(data-value)`
  - specifies or returns the JVM token, a fullword binary value that identifies the JVM. This information is returned by an INQUIRE JVM browse, and you can then specify the JVM token for the JVM about which you want to inquire. The JVM token 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.

For more information about the use of CVDAs, see [“CICS-value data areas (CVDAs)” on page 8](#).
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.

CLASSCACHEST(cvda)
returns a CVDA value indicating whether the JVM is a worker JVM dependent on the shared class cache. CVDA values are:

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. CVDA values are:

CICSEXECKEY
The JVM executes in CICS key.

USEREXECKEY
The JVM executes in user key.

PHASINGOUTST(cvda)
returns a CVDA value indicating whether the JVM is being phased out as a result of an EXEC CICS SET JVMPOOL TERMINATE or EXEC CICS PERFORM CLASSCACHE TERMINATE command (or the equivalent CEMT commands). CVDA values are:

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 the level of reusability for this JVM. CVDA values are:

RESET
The JVM profile for this JVM specified that it is resettable (or in the case of a worker JVM, the JVM profile for the master JVM specified that the master and worker JVMs are resettable).

REUSE
The JVM profile for this JVM specified that it is continuous (or in the case of a worker JVM, the JVM profile for the master JVM specified that the master and worker JVMs are continuous).

NOREUSE
The JVM profile for this JVM specified that it is single-use.
INQUIRE JVM

**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 has been given when a browse of JVMs 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:
4 The named JVM was not found
INQUIRE JVMPOOL

Retrieve information about the pool of JVMs in the CICS address space.

**Description**

The INQUIRE JVMPOOL command returns information about the pool of JVMs in the CICS region. There can be only one pool of JVMs in a CICS region, therefore there is no name or identifier required on this command.

CICS determines the information you request from the actual JVMs that are started and pre-initialized in the CICS region.

**Options**

- `JVMLVL0TRACE(data-area)`
  - returns a 240-character data value giving the current options for JVM tracing, which are used when the associated level of tracing has been activated using the CICS-supplied transaction CETR, the SET TRACETYPE command, or the system initialization parameter SPCTRSJ. Trace levels 29–32 for the SJ component correspond to JVMLVL0TRACE, JVMLVL1TRACE, JVMLVL2TRACE and JVMUSERTRACE respectively.

- `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 executing, or has executed, a Java program in the JVM.

When you issue an EXEC CICS SET JVMPOOL TERMINATE command (or the equivalent CEMT command), all the JVMs in the JVM pool are marked for removal. When you issue an EXEC CICS PERFORM CLASSCACHE TERMINATE command (or the equivalent CEMT command), all worker JVMs using the active shared class cache are marked for removal.

- `STATUS(cvda)`
  - returns a CVDA indicating the overall status of the JVM pool. The CVDA values are:

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
INQUIRE JVMPOOL

ENABLED
The pool is enabled for use and Java programs can execute using JVMs from the pool. This is the normal status.

DISABLED
The pool is disabled, and new requests cannot be serviced from the pool. Programs can still be executing if they were started before the JVM pool became 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, and report their full path name and association with the shared class cache.

Description

The INQUIRE JVMPROFILE command retrieves the full path name of the HFS file for a JVM profile, and tells you whether or not a JVM with this profile uses the shared class cache.

INQUIRE JVMPROFILE can only find JVM profiles that have been used during the lifetime of this CICS region, for JVMs that can run applications. It does not find JVM profiles that have not been used, or JVM profiles that have only been used for the master JVM that initializes the shared class cache.

Browsing

You can browse through all of the JVM profiles that have been used in a CICS region by using the browse options (START, NEXT, and END) on INQUIRE JVMPROFILE commands. Note that INQUIRE JVMPROFILE commands do not support the AT option for browsing. In browse mode, the JVM profiles are returned in alphabetical order. See "Browsing resource definitions" on page 21 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 definition.

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 HFS file name.

HFSNAME(data-area)

returns a 255–character area containing the full path name of the HFS file for the JVM profile.

CLASSCACHEST(cvda)

returns a CVDA value indicating whether JVMs that use this JVM profile are worker JVMs dependent on the shared class cache. CVDA values are:

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
INQUIRE JVMPROFILE

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 the level of reusability for JVMs that are
   created using this JVM profile. CVDA values are:

   RESET
      JVMs with this JVM profile are resettable.

   REUSE
      JVMs with this JVM profile are continuous.

   NOREUSE
      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 MODENAME

Retrieve information about a session group within a connection.

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-area for MODENAME is optional; if you provide it, CICS uses it to return the name of the session group. The data-area for CONNECTION is required, however. If you want to limit your browse to a single connection, specify its name there. To see all groups, initialize this value to nulls on each INQUIRE MODENAME NEXT command, and CICS will use the data-area to return the connection name.

See “Browsing resource definitions” on page 21 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.

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.
INQUIRE MODENAME

**AUTOCONNECT**(cvda)
returns a CVDA value indicating whether the sessions within this group are to be bound automatically whenever CICS starts communication with VTAM. 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. This 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.
The connection specified on an INQUIRE MODENAME NEXT cannot be found.
INQUIRE MONITOR

Retrieve the status of CICS monitoring.

Description

The INQUIRE MONITOR command allows you to find out whether CICS monitoring is active, what types of data are being recorded, and other recording options.

CICS monitoring is controlled by a master switch (the STATUS option) and two switches that govern what types of data are recorded (the EXCEPTCLASS and PERFCLASS options). See the SET MONITOR command on page 553 for a description of monitor data classes and details of how the switches interact.

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.
Options

`APPLNAMEST(cvda)`
returns a CVDA value indicating whether CICS application naming support is enabled. CVDA values are:

`APPLNAME`
CICS application naming support is enabled.

`NOAPPLNAME`
CICS application naming support is not enabled.

`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:

`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.

`EXCEPTCLASS(cvda)`
returns a CVDA value indicating whether the exception class of monitoring data is recorded when monitoring is active. CVDA values are:

`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.

There are two formats for the frequency interval:

- 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 means 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).
INQUIRE MONITOR

FREQUENCYSEC(data-area)
returns the seconds component of the frequency interval, in fullword binary form (see the FREQUENCY option).

PERFCLASS(cvda)
returns a CVDA value indicating whether the performance class of monitoring data is recorded when monitoring is active. CVDA values are:

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:

NORESRC  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:

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:

OFF     CICS monitoring is not active in the region. No monitoring data is accumulated or written out, irrespective of the settings of the monitoring data classes.
ON      CICS monitoring is active. Data is accumulated for all classes of monitor data, and written out for those classes that are active.

SUBSYSTEMID(data-area)
returns the 4-character name used as the subsystem identification in the MVS workload activity reports. For information on the MVS workload activity reports and the subsystem identification, see the CICS Performance Guide. For information on how to specify the subsystem identification, see the MNSUBSYS parameter in the CICS System Definition Guide.

SYNCPOINTST(cvda)
returns a CVDA value indicating whether CICS records performance class data separately for each unit of work (UOW) within tasks that contain multiple UOWs. A UOW within a task ends when a syncpoint 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 syncpoint, the UOW does not end. CVDA values are:

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:

- 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 MVSTCB

Retrieve addresses and storage usage information for MVS TCBs.

**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 MVSTCB command returns the address and storage usage information for the MVS TCB.

**Browsing**

This command can only be used in browse mode.

**Options**

**ELEMENTLIST(ptr-ref)**

returns the address of a list of the addresses of all areas of private storage owned by this TCB. The number of addresses in this list is the NUMELEMENTS option value.

CICS obtains the storage for this list and frees it when the inquiring task ends, or issues another INQUIRE MVSTCB command with the NUMELEMENTS option. The task cannot free the storage itself.

**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.

CICS obtains the storage for this list and frees it when the inquiring task ends, or issues another INQUIRE MVSTCB command with the NUMELEMENTS option. The task cannot free the storage itself.

**NUMELEMENTS(data-area)**

returns the number of storage elements, in fullword binary form, owned by this TCB, in the data area you provide. If you request an ELEMENTLIST, LENGTHLIST, or SUBPOOLLIST, this value is the number of entries in the list.

**MVSTCB(data-area)**

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 the storage and CPU time
information for the TCB.

SUBPOOLLIST(ptr-ref)
returns the address of a list of fullword binary subpool numbers. Each entry in
this list is the number of the MVS subpool for the corresponding entry in the
ELEMENTLIST list.

CICS obtains the storage for this list and frees it when the inquiring task ends,
or issues another INQUIRE MVSTCB command with the NUMELEMENTS
option. The task cannot free the storage itself.

Conditions

END
RESP2 values:
2 All authorised 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

INQUIRE NETNAME
Retrieve information about a terminal or session.

INQUIRE NETNAME/TERMINAL

```
INQUIRE NETNAME(data-value)  | options
  TERMINAL(data-area)        |
```

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 VTAM. For ISC sessions, it is the name by which the session (or session group, if there are parallel sessions) is known to VTAM. 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. VTAM 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 21 for general information about browsing, including syntax, exception conditions, and examples.
INQUIRE NETNAME

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 358 for details of the options and conditions that apply to the INQUIRE NETNAME command.
INQUIRE PARTNER

Retrieve information about a partner.

---

INQUIRE PARTNER

- `<INQUIRE PARTNER(data-value)>
  - NETWORK(data-area)`
- `<INQUIRE PARTNER(data-value)>
  - NETNAME(data-area)`
- `<INQUIRE PARTNER(data-value)>
  - PROFILE(data-area)`
- `<INQUIRE PARTNER(data-value)>
  - TPNAME(data-area)`
- `<INQUIRE PARTNER(data-value)>
  - TPNAMELEN(data-area)`

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 21 for general information about browsing, including syntax, exception conditions, and examples.

Options

- **NETNAME(data-area)**
  returns the 8-character name of the VTAM 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.
INQUIRE PARTNER

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.

#### 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” on page 21 for general information about browsing, including syntax, exception conditions, and examples.

#### Options

**PIPELINE**(data-value)

Specifies the name of the PIPELINE about which you are inquiring. The name can be up to 8 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.

**ENABLESTATUS**(cvda)

Returns the status of the PIPELINE:

- **ENABLED**
  
  The PIPELINE is ready for use.

- **DISABLED**
  
  The PIPELINE is not processing requests, and is unable to accept new work. It may have failed to initialize, or may 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 currently-executing work to complete.

---

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.
INQUIRE PIPELINE

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 currently-executing work to complete.

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.

SHELF(data-area)
Returns the name of the shelf directory. The name can be up to 255 characters long.

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

INQUIRE PROCESSTYPE

Browsing

Retrieve the attributes of a CICS business transaction services (BTS) process-type.

INQUIRE PROCESSTYPE

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 21 for general information about browsing, including syntax, exception conditions, and examples.

Options

AUDITLEVEL(cvda)
indicates the level of audit currently active for processes of the specified type.
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 manual.

AUDITLOG(data-area)
returns the 8-character name of the CICS journal used as the audit log for processes of the specified type.

FILE(data-area)
returns the 8-character name of the CICS file associated with the process-type.
INQUIRE PROCESSTYPE

**PROCESSTYPE**(data-value)

specifies the name (1–8 characters) of the process-type being inquired upon.

**STATUS**(cvda)

indicates whether new processes of the specified type can currently be defined.

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**

**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 could not be found.
INQUIRE PROFILE

Determine whether a transaction profile is installed.

INQUIRE PROFILE

> INQUIRE PROFILE(data-value)

Conditions: END, ILLOGIC, NOTAUTH, PROFILEIDERR

Description

The INQUIRE PROFILE command allows you 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 21 for general information about browsing, including syntax, exception conditions, and examples.

Options

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.

**Description**

The INQUIRE PROGRAM command returns information about 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 INQUIRE command for all three. To avoid confusion, we use the word *module* to refer to the object of your inquiry, except in some cases where the option applies only to executable programs.

CICS determines the information 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

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
INQUIRE PROGRAM

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
alphabetical order, and you can specify a starting point with the AT option if you
wish. See "Browsing resource definitions" on page 21 for general information about
browsing, including syntax, exception conditions, and examples.

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. CVDA values are:

CICSAPI

The program is restricted to use of the CICS permitted application
programming interfaces only. Dependent upon the program's
CONCURRENCY setting, the application will either always run on the
quasi-reentrant (QR) TCB, or if it is defined as threadsafe it may run on
whichever TCB in use by CICS at the time which is determined as
suitable.

OPENAPI

The program is not restricted to the CICS permitted application program
interfaces only. CICS will execute the program on its own L8 or L9
mode open TCB dependent upon the EXECKEY setting. If when
executing a CICS command, CICS requires a switch to QR TCB, it will
return to the open TCB before handing control back to the application
program. OPENAPI requires the program to be coded to threadsafe
standards and defined with CONCURRENCY(THREADSAFE).

Notes:

1. The API 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.

2. You cannot modify a program's API attribute using the SPI--the API option is
not supported on the EXEC CICS SET PROGRAM command. You can only
change the API attribute by redefining the program's API option in the CICS
program resource definition, or in the program autoinstall model, and
reinstalling the definition.

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:

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 is not applicable because the module is a remote program, a map set, or a partition set.

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:

COBOL
The module is an OS/VS COBOL program. (OS/VS COBOL programs cannot run under CICS Transaction Server for z/OS Version 3.)

COBOLII
The module is a VS COBOL II program.

NOTAPPLIC
COBOL type does not apply, because the module is a remote program, a map set, or a partition set. This CVDA value is always returned for JVM programs.

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 indicating the concurrency attribute of the installed program definition. The CVDA values are:

QUASIRENT
The program is defined as being quasi-reentrant, and is able to run only under the CICS QR TCB.

THREADSAFE
The program is defined as threadsafe, and is able to run under whichever TCB is in use by its user task when the program is given control. This could be either an open TCB or the CICS QR TCB.

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 subsequently loaded. If CICS finds that the program itself contains a CONCURRENCY value defined by Language Environment run-time options, the installed program resource definition is updated by the Language Environment run-time 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 a program's concurrency attribute using the SPI—the CONCURRENCY option is not supported on the EXEC CICS SET PROGRAM command. You can only change the concurrency by redefining the program's CONCURRENCY option 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, you need to issue a SET PROGRAM COPY command or its CEMT equivalent. You should ensure that the program exists in one of the libraries in the DFHRPL concatenation before doing so. CVDA values are:

NOTREQUIRED
A new copy is not required. This CVDA value is always returned for JVM programs.

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:

ANY
The program can accept an address above 16MB.

BELOW
The program requires any data address returned to it from CICS to be less than 16MB.

NOTAPPLIC
The option is not applicable because the module is a remote program, a map set, or a partition set.

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:

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 execute.

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 execute 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.

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 JVM program, 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:

**CICSEXECKEY**
The program executes in CICS key.

**NOTAPPLIC**
The module is a remote program, a map set, or a partition set.

**USEREXECKEY**
The program executes 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. EXECUTIONSET applies only to executable programs, and governs the API only when a program is invoked locally. (When it is invoked 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:

**DPLSUBSET**
The program is always restricted.

**FULLAPI**
The program is not restricted unless invoked 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:

**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.

**JVMCLASS(data-area) (JVM programs only)**
returns the name, in 255 characters, of any class specified in the program definition.

**JVMDEBUG(cvda) (JVM programs only)**
This option is obsolete. CICS always returns NODEBUG as the cvda value.

**JVMPROFILE(data-area) (JVM programs only)**
returns the 8-character name of the JVM profile associated with the program.

**LANGDEPRECATED(cvda) (programs only)**
returns a CVDA value indicating the language (that is, the module is loaded). 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:

**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 where programs are written in assembler and do not have the DFHEAI stub. In this case, the LANGUAGE attribute of the program definition is used to return a value. CVDA values are:

**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 JVM program.

**LOADPOINT(ptr-ref)**
returns the load address of the module. If it is not currently loaded, or if the program is running under a JVM, a null pointer (X'FF000000') is returned.
INQUIRE PROGRAM

LPASTATUS(cvda)
returns a CVDA value indicating whether the module resided in the link pack area when it was last used. CVDA values are:

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 JVM program.

NOTLPA The copy used was in CICS dynamic storage.

PROGRAM(data-value)
specifies the 8-character name of the program, map set, or partition set about which you are inquiring.

PROGTYPE(cvda)
returns a CVDA value indicating the type of module. CVDA values are:

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. REMOTENAME 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 JVM program.

RUNTIME(cvda)
returns a CVDA value indicating the runtime environment of the program. CVDA values are:

JVM The program is a Java program that will run under the control of a Java Virtual Machine.

LE370 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.
INQUIRE PROGRAM

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 should obtain the module the next time a new copy is required. CVDA values are:

NOTAPPLIC SHARESTATUS is not applicable because the module is a remote program or a JVM program.

PRIVATE The module is loaded from the concatenated libraries named on the DFHRPL DD statement.

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:

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, executes remotely (that is, the transaction identifier 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. A value of -1 is returned if the program is remote, or a JVM program.

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

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

PGMIDERR values:
1 The program cannot be found. If this error occurs on an INQUIRE PROGRAM NEXT, 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.

#### 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.

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.

#### Conditions

- END
- ILLOGIC
- INVREQ
- NOTAUTH
- NOTFND

---

**INQUIRE REQID**

<table>
<thead>
<tr>
<th>Command: INQUIRE REQID(data-value)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>REQTYPE(cvda)</strong></td>
</tr>
<tr>
<td><strong>TERMID(data-area)</strong></td>
</tr>
<tr>
<td><strong>TRANSID(data-area)</strong></td>
</tr>
<tr>
<td><strong>INTERVAL(data-area)</strong></td>
</tr>
<tr>
<td><strong>TIME(data-area)</strong></td>
</tr>
<tr>
<td><strong>AFTER hours minutes seconds</strong></td>
</tr>
<tr>
<td><strong>AT hours minutes seconds</strong></td>
</tr>
<tr>
<td><strong>SET(ptr-ref) LENGTH(data-area)</strong></td>
</tr>
<tr>
<td><strong>FMHSTATUS(cvda)</strong></td>
</tr>
<tr>
<td><strong>RTRANSID(data-area)</strong></td>
</tr>
<tr>
<td><strong>RTERMID(data-area)</strong></td>
</tr>
<tr>
<td><strong>QUEUE(data-area)</strong></td>
</tr>
<tr>
<td><strong>USERID(data-area)</strong></td>
</tr>
<tr>
<td><strong>HOURS(data-area)</strong></td>
</tr>
<tr>
<td><strong>MINUTES(data-area)</strong></td>
</tr>
<tr>
<td><strong>SECONDS(data-area)</strong></td>
</tr>
</tbody>
</table>

**Conditions:** END, ILLOGIC, INVREQ, NOTAUTH, NOTFND
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 (0hhmmss+), 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 21 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:
INQUIRE REQID

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 0hhmmss+.

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).

REQTYP(*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:
INQUIRE REQID

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:

1 The REQID cannot be found.
INQUIRE REQUESTMODEL

Retrieve information about a REQUESTMODEL.

**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 21 for general information about browsing, including syntax, exception conditions, and examples. You can specify the following options on the INQUIRE request:

**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.

- **CORBASERVER**(data-area)**
  returns a 4-character field containing the (possibly generic) name of the destination CorbaServer for this REQUESTMODEL.

- **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:
INQUIRE REQUESTMODEL

HOME specifies that this is the home interface for the bean.
REMOTE specifies that this 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 executed 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:
  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 an 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

\[\text{INQUIRE RRMS} \quad \text{OPENSTATUS}(cvda)\]

**Conditions:** NOTAUTH

**Description**

The INQUIRE RRMS command indicates whether inbound transactional EXCI work is currently being accepted.

**Options**

\[\text{OPENSTATUS}(cvda)\]

- 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.

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 \texttt{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 \texttt{ENDOFDAY}, \texttt{INTERVAL}, and \texttt{NEXTTIME} options.
- Separate hours, minutes, and seconds, which you obtain by specifying the \texttt{ENDOFDAYHRS}, \texttt{ENDOFDAYMINS}, and \texttt{ENDOFDAYSECS} options (instead of \texttt{ENDOFDAY}), \texttt{INTERVALHRS}, \texttt{INTERVALMINS}, and \texttt{INTERVALSECS} (instead of \texttt{INTERVAL}) and \texttt{NEXTTIMEHRS}, \texttt{NEXTTIMEMINS}, and \texttt{NEXTTIMESECS} (instead of \texttt{NEXTTIME}).

The [CICS Performance Guide](#) contains more detail about CICS statistics, and the \texttt{SET STATISTICS} command on page 566 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 \texttt{RECORDING} switch is not changed from its current value). This is the end-of-day time if \texttt{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.
INQUIRE STATISTICS

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 command, described on page 466 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.

**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.

**Conditions:** NOTAUTH, TASKIDERR
INQUIRE STORAGE

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

Retrieves information about a currently connected MVS log stream.

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 21 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.
INQUIRE STREAMNAME

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.

If the log stream name refers to the CICS system log, the use count is always 1. This is so, even when user application programs write recovery records to the CICS system log.

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.
2 The browse token is not valid.

NOTAUTH
RESP2 values:
100 The user is not authorized for this command.

NOTFND
RESP2 values:
1 The requested log stream name was not found.
INQUIRE SUBPOOL

Retrieve information about storage subpools in the CICS region.

INQUIRE SUBPOOL

INQUIRE SUBPOOL(data-area)

DSANAME(data-area)

SUBPOOL(data-area)

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 SUBPOOL command returns information about a particular storage subpool.

Browsing

You can also browse through all the storage subpools in the region using the browse options (START, AT, NEXT, and END) on INQUIRE SUBPOOL command. In browse mode, the definitions are returned in alphabetic order of subpool name. You can specify a starting point anywhere in the full range of subpools using the AT option. If you want to see all the subpools with names beginning with a certain string of characters, for example, you can start your browse with an AT value comprising those characters, padded on the right with nulls (X'00') up make up the eight characters.

See "Browsing resource definitions" on page 21 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
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 the CICS Performance Guide.

For browse operations, specify SUBPOOL on the START browse request only, not on the NEXT or END requests.
INQUIRE SUBOOL

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

Retrieves information about a system dump table entry.

**Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURRENT(data-area)</td>
<td>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)</td>
</tr>
</tbody>
</table>

**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 SYSDUMPCODE command allows you to look at some of the information in a system dump code table entry.

The table entry tells CICS which actions to take when a system dump request with this code occurs, and how many times to take them (the MAXIMUM option); requests received after the maximum are counted (the CURRENT option), but are otherwise ignored.

CICS provides a system dump table with entries for some CICS-defined system dump codes. If it receives a dump request for a code for which it does not have an entry, it builds one, using default values. You can add your own entries with the SET SYSDUMPCODE command or a CEMT transaction. Entries you add remain over executions of CICS until an initial or cold start occurs, but the entries that CICS builds are considered to be 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 system dump code table by using the browse options (START, NEXT, and END) on INQUIRE SYSDUMPCODE commands. See "Browsing resource definitions" on page 21 for general information about browsing, including syntax, exception conditions, and examples.
INQUIRE SYSDUMPCODE

SYSDUMPCODE command or its CEMT equivalent.) The count includes requests that do not result in a dump because either CICS or MVS suppressed it.

DAEOPTION
returns a CVDA value identifying whether a dump produced for this dump code is eligible for subsequent 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, 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 the z/OS V1R4.0 MVS Diagnosis: Tools and Service Aids manual.

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 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 SYSUMP), 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.

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
INQUIRE SYSDUMPCODE

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 on page [INQUIRE SYSTEM command options - DTRPROGRAM” on page 321]. 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

ACTOPENCBS(data-area)  MAXOPENCBS(data-area)
AKP(data-area)  MAXTASKS(data-area)
CDSASIZE(data-area)  MRBATCH(data-area)
CICSSTATUS(cvda)  OPREL(data-area)
CICSSYS(data-area)  OPSYS(data-area)
CICSTLEVEL(data-area)  OSLEVEL(data-area)
CMOPROTECT(cvda)  PROGAUTOCTLG(cvda)
COLDSTATUS(cvda)  PROGAUTOEXIT(data-area)
DB2CONN(data-area)  PROGAUTOINST(cvda)
DEBUGTOOL(cvda)  PRTYAGING(data-area)
DFTUSER(data-area)  ROSASIZE(data-area)
DSALIMIT(data-area)  REENTPROTECT(cvda)
DSRTPROGRAM(data-area)  RELEASE(data-area)
DTRPROGRAM(data-area)  RLSTSTATUS(cvda)
DUMPING(cvda)  RUNAWAY(data-area)
EDSASIZE(data-area)  SCANDELAY(data-area)
EDSALIMIT(data-area)  SDSASIZE(data-area)
ERDSASIZE(data-area)  SDTRAN(data-area)
ERDSASIZE(data-area)  SECURITYMGR(cvda)
EUDSASIZE(data-area)  SHUTSTATUS(cvda)
FORCQR(cvda)  SOSSTATUS(cvda)
GMTTEXT(data-area)  STARTUP(cvda)
GMLLENGTH(data-area)  STARTUPDATE(data-area)
GMTRANDID(data-area)  STOREPROTECT(cvda)
INITSTATUS(cvda)  TIME(data-area)
JOBNAM(data-area)  TRANISOLATE(cvda)
LOGDEFER(data-area)  UDASIZE(data-area)
XRFSTATUS(cvda)  XRFSTATUS(cvda)

Condition: NOTAUTH

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.

Description

The INQUIRE SYSTEM command returns information about the CICS system under which the task issuing the command is executing.

Many of the options in this command correspond to system initialization parameters and take their initial values from the parameters. Some of these 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 still others return...
INQUIRE SYSTEM

information determined solely by the current state of the system. Table 3 indicates where the option values come from and, in the case of those specified in the system initialization parameter, the name of the parameter. For these options, the **CICS System Definition Guide** is a good source of additional information. For state options, the **CICS Customization Guide** is the primary source.

Table 3. INQUIRE SYSTEM options

<table>
<thead>
<tr>
<th>Option</th>
<th>Origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTOPENTCBS</td>
<td>System state</td>
</tr>
<tr>
<td>AKP</td>
<td>AKPFREQ system initialization parameter</td>
</tr>
<tr>
<td>CDSASIZE</td>
<td>System state</td>
</tr>
<tr>
<td>CICSSTATUS</td>
<td>System state</td>
</tr>
<tr>
<td>CICSSYS</td>
<td>System state</td>
</tr>
<tr>
<td>CICSTSLEVEL</td>
<td>CICS control block</td>
</tr>
<tr>
<td>CMDPROTECT</td>
<td>CMDPROT system initialization parameter</td>
</tr>
<tr>
<td>COLDSTATUS</td>
<td>System state</td>
</tr>
<tr>
<td>DB2CONN</td>
<td>DB2CONN system initialization parameter</td>
</tr>
<tr>
<td>DEBUGTOOL</td>
<td>DEBUGTOOL system initialization parameter</td>
</tr>
<tr>
<td>DFLTUSER</td>
<td>DFLTUSER system initialization parameter</td>
</tr>
<tr>
<td>DSRTPROGRAM</td>
<td>DSRTPGM system initialization parameter</td>
</tr>
<tr>
<td>DTRPROGRAM</td>
<td>DTRPGM system initialization parameter</td>
</tr>
<tr>
<td>DUMPING</td>
<td>DUMP system initialization parameter</td>
</tr>
<tr>
<td>ECDSASIZE</td>
<td>System state</td>
</tr>
<tr>
<td>EDSALIMIT</td>
<td>EDSALIM system initialization parameter</td>
</tr>
<tr>
<td>ERDSASIZE</td>
<td>System state</td>
</tr>
<tr>
<td>ESDASIZE</td>
<td>System state</td>
</tr>
<tr>
<td>EUDSASIZE</td>
<td>System state</td>
</tr>
<tr>
<td>FORCEQR</td>
<td>FORCEQR system initialization parameter</td>
</tr>
<tr>
<td>GMMTEXT, GMMLENGTH</td>
<td>GMTEXT system initialization parameter</td>
</tr>
<tr>
<td>GMMTRANID</td>
<td>GMTRAN system initialization parameter</td>
</tr>
<tr>
<td>INITSTATUS</td>
<td>System state</td>
</tr>
<tr>
<td>JOBNAME</td>
<td>JCL or cataloged procedure</td>
</tr>
<tr>
<td>LOGDEFER</td>
<td>LGDFINT system initialization parameter</td>
</tr>
<tr>
<td>MAXXOPENTCBS</td>
<td>MAXXOPENTCBS system initialization parameter</td>
</tr>
<tr>
<td>MAXTASKS</td>
<td>MXT system initialization parameter</td>
</tr>
<tr>
<td>MROBATCH</td>
<td>MROBATCH system initialization parameter</td>
</tr>
<tr>
<td>OPREL</td>
<td>Operating system (MVS)</td>
</tr>
<tr>
<td>OPSYS</td>
<td>Operating system (MVS)</td>
</tr>
<tr>
<td>OLEVEL</td>
<td>Operating system (OS/390)</td>
</tr>
<tr>
<td>PROGAUTOCTLG</td>
<td>PGAICTLG system initialization parameter</td>
</tr>
<tr>
<td>PROGAUTOEXIT</td>
<td>PGAIEXIT system initialization parameter</td>
</tr>
<tr>
<td>PROGAUTOINST</td>
<td>PGAIPGM system initialization parameter</td>
</tr>
<tr>
<td>PRTYAGING</td>
<td>PRTYAGE system initialization parameter</td>
</tr>
<tr>
<td>RDSASIZE</td>
<td>System state</td>
</tr>
<tr>
<td>REENTPROTECT</td>
<td>RENTPGM system initialization parameter</td>
</tr>
<tr>
<td>RELEASE</td>
<td>CICS system code</td>
</tr>
<tr>
<td>RLSSTATUS</td>
<td>RLS system initialization parameter</td>
</tr>
<tr>
<td>RUNAWAY</td>
<td>ICVR system initialization parameter</td>
</tr>
<tr>
<td>SCANDELAY</td>
<td>ICVTSO system initialization parameter</td>
</tr>
<tr>
<td>SDSASIZE</td>
<td>System state</td>
</tr>
<tr>
<td>SDTRAN</td>
<td>SDTRAN system initialization parameter</td>
</tr>
<tr>
<td>SECURITYMGR</td>
<td>SEC system initialization parameter</td>
</tr>
<tr>
<td>SHUTSTATUS</td>
<td>System state</td>
</tr>
<tr>
<td>SOSSTATUS</td>
<td>System state</td>
</tr>
</tbody>
</table>
Table 3. INQUIRE SYSTEM options (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARTUP</td>
<td>System state</td>
</tr>
<tr>
<td>STARTUPDATE</td>
<td>System state</td>
</tr>
<tr>
<td>STOREPROTECT</td>
<td>STGPROT system initialization parameter, and hardware</td>
</tr>
<tr>
<td>TIME</td>
<td>ICV system initialization parameter</td>
</tr>
<tr>
<td>TRANISOLATE</td>
<td>TRANISO system initialization parameter, and hardware</td>
</tr>
<tr>
<td>UDSASIZE</td>
<td>System state</td>
</tr>
<tr>
<td>XRFSTATUS</td>
<td>XRF system initialization parameter, and system state</td>
</tr>
</tbody>
</table>

**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 it 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 overall limit for dynamic storage areas that reside below 16MB (the DSALIMIT option value).

**CICSSTATUS**(cvda)

returns a CVDA value identifying the current execution status of CICS. CVDA values are:

**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.
INQUIRE SYSTEM

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 on page "INQUIRE SYSTEM command options - GMMTRANID" on page 322 for further information.

CICSSYS(data-area)
returns a 1-character value identifying the operating system for which the running CICS system has been built. A value of “X” represents MVS/ESA.

CICSTSLEVEL(data-area)
returns a 6-character value identifying the version, release, and modification level of the CICS Transaction Server for OS/390® product under which the CICS region is running. The value is of the form vvrrmm, and CICS Transaction Server for z/OS Version 3 Release 1 returns 030100.

CMDPROTECT(cvda)
returns a CVDA value indicating whether command protection is active or not. With command protection active, when a task issues a command, 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:

CMDPROT
Command protection is active.

NOCMDPROT
Command protection is not active.

COLDSTATUS(cvda)
returns a CVDA value indicating 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:

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 currently installed, or blanks if no DB2CONN is currently installed. DB2CONN allows the user to determine the name of the RDO DB2CONN definition. Only one DB2CONN can be installed at a time. A DB2CONN defines the global attributes of the connection to be established between DB2 and CICS.

DEBUGTOOL(cvda)
returns a CVDA value indicating 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, such as the CICS Execution Diagnostic Facility
(CEDF) do not use debugging profiles.

The CVDA values are:

**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 the 16MB boundary. (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 indicating whether the taking of CICS system dumps is
suppressed. CVDA values are:

**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 overall limit for dynamic storage areas that reside above the 16MB
boundary (the EDSALIMIT option value).

**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 four individual
dynamic storage areas that reside above the 16MB boundary. (See the
ECDSASIZE, ERDSASIZE, ESDSASIZE, and EUDSASIZE options of this
command.)
INQUIRE SYSTEM

**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 overall limit for dynamic storage areas that reside above the 16MB boundary (the EDSALIMIT option value).

**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 overall limit for dynamic storage areas that reside above the 16MB boundary (the EDSALIMIT option value).

**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 overall limit for dynamic storage areas that reside above the 16MB boundary (the EDSALIMIT option value).

**FORCEQR(cvda)**
returns a CVDA value indicating whether CICS is to force CICSAPI user application programs that are defined as threadsafe to run on the quasi-reentrant (QR) TCB. The CVDA values are:

- **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 OPENAPI programs or C/C++ programs compiled with XPLINK.

- **NOFORCE**
  CICS is to honor the CONCURRENCY(THREADSAFE) attribute on CICSAPI user application programs, and allow user programs to run on an open TCB to avoid unnecessary TCB switching.

**GMMLENGTH(data-area)**
returns a halfword binary field giving 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 giving the initialization status of the CICS system. CVDA values are:

- **FIRSTINIT**
  First stage of CICS initialization.

- **INITCOMPLETE**
  CICS initialization is complete.
SECONDINIT
Second stage of initialization.

THIRDINIT
Third stage of initialization.

See the CICS Customization Guide for more information about CICS initialization.

JOBNAME(data-area)
returns the 8-character MVS jobname under which CICS is running.

LOGDEFER(data-area)
returns the halfword binary value giving 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 the CICS System Definition Guide for information about the LOGDEFER parameter and associated system initialization parameter LGDFINT.

MAXOPENTCBS(data-area)
returns a fullword binary field giving the maximum number of open TCBs that CICS is allowed to attach and maintain in its pool of open TCBs. For information about the number actually 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 giving 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 in CICS Transaction Server for z/OS either, although they did in earlier releases.

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.

OPREL(data-area) (supported for compatibility only)
returns a halfword binary field giving the last 2 digits of the level number of the operating system under which the CICS region is running. For example, OS/390 Release 4 is represented by 04.

Note: This field is supported for compatibility purposes only. The information is derived from the last two numbers held in the MVS CVTPRODN field. For example, CVTPRODN holds the value SP5.2.2 for MVS/ESA SP™ Version 5 Release 2.2 (in which case OPREL returns 22), and SP6.0.3 for OS/390 Release 3. See the OSLEVEL field for the full version and release number of OS/390.

OPSYS(data-area)
returns a 1-character value identifying the operating system under which CICS is running. A value of “X” represents MVS/ESA.

OSLEVEL(data-area)
returns a 6-byte field containing the version, release, and modification level of the OS/390 product on which CICS is running. For example, OS/390 Version 2 Release 4 Modification 0 returns the value 020400.
INQUIRE SYSTEM

PROGAUTOCTLG(cvda)
returns a CVDA value indicating 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:

**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 indicating 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:

**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
overall limit for dynamic storage areas that reside below the 16MB boundary
(the DSALIMIT option value).

REENTPROTECT(cvda)
returns a CVDA value indicating 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:

**REENTPROT**
Read-only DSAs are in key 0 storage.

**NOREENTPROT**
Read-only DSAs are in CICS-key storage.
**INQUIRE SYSTEM**

**RELEASE**(*data-area*)  *(supported for compatibility only)*
returns a 4-character string comprising the value 640, indicating the level number of the CICS code within the CICS Transaction Server for OS/390.

This option is supported only for compatibility with earlier releases. As an exclusive element of CICS Transaction Server for OS/390, 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 indicating whether VSAM RLS is active—that is, the CICS region is registered (with a currently-open control ACB) with an SMSVSAM address space. CVDA values are:

- **NOTAPPLIC**
  This CICS region does not support VSAM RLS because:
  - CICS initialized with RLS=NO as a system initialization parameter, or
  - 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 giving 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 on page 395).

**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 overall limit for dynamic storage areas that reside below the 16MB boundary (the DSALIMIT option value).

**SDTRAN**(*data-area*)
returns the 4-character name of the transaction to be run at the beginning of normal or immediate shutdown. This may be the name of a user-supplied transaction, or the CICS-supplied default transaction, CESD.

**SECURITYMGR**(*cvda*)
returns a CVDA value identifying whether an external security manager (such as RACF) is active in the system, or whether no security is being used. CVDA values are:
INQUIRE SYSTEM

EXTSECURITY
An external security manager is active.

NOSECURITY
No security is being used.

SHUTSTATUS(cvda)
returns a CVDA value indicating the shutdown status of CICS (see the
CICSSTATUS option on page “CICSSTATUS” on page 319). CVDA values are:

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.

SOSSTATUS(cvda)
returns a CVDA value indicating whether CICS is short on storage. CVDA
values are:

NOTSOS
CICS is not short on storage in any of the dynamic storage areas.

SOS
CICS is short on storage in at least one dynamic storage area above
and at least one below the 16MB line.

SOSABOVE
CICS is short on storage in at least one dynamic storage area above
16MB, but none below.

SOSBELOW
CICS is short on storage in at least one dynamic storage area below
16MB, but none above.

STARTUP(cvda)
returns a CVDA value indicating how the current execution of CICS started.
CVDA values are:

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 indicating whether storage protection is active or not. For storage protection to be active, it must be requested (the STGPROT option in the system initialization table), and it must be supported by the hardware. CVDA values are:

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 indicating whether transaction isolation is active or not. For it to be active, both transaction isolation and storage protection must be requested (the TRANISO and STGPROT system initialization parameters), and it must be supported by the hardware. CVDA values are:

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 overall limit for dynamic storage areas that reside below the 16MB boundary (the DSALIMIT option value).

XRFSTATUS(cvda)
returns a CVDA value indicating whether the current execution of CICS started as an active or alternate region under the extended recovery facility (XRF). CVDA values are:

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.

**Conditions:** INVREQ, NOTAUTH, TASKIDERR

This command is threadsafe.
For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
INQUIRE TASK

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 CICS business transaction services activity that this task is executing on behalf of.

ACTIVITYID(data-area)
returns the 52-character, CICS-assigned, identifier of the CICS business transaction services 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\((data\text{-}area)\)
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:

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 SECURITYMGK option of the INQUIRE SYSTEM command on page \[325\] and the CICS Resource Definition Guide for more information).

CURRENTPROG\((data\text{-}area)\)
returns a 1- to 8-character name of the current program, as known to the CICS program manager domain, executing for this this task.

DB2PLAN\((data\text{-}area)\)
returns a 1- to 8-character name of the DB2PLAN being used by this task, or blanks if no DB2PLAN is being used.

DTIMEOUT\((data\text{-}area)\)
returns a fullword binary field giving the deadlock time-out interval, in seconds. CICS abends a task that waits longer than its deadlock time-out 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:

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\((data\text{-}area)\)
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:
INQUIRE TASK

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 TERMID 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 TERMID.

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 MQ™ correlator for the 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 in-doubt 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 in-doubt 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 in-doubt 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 in-doubt state. CVDA values are:

   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.
INQUIRE TASK

For further information about the meaning of the ACTION and WAIT attributes of the TRANSACTION definition, see the CICS Resource Definition Guide.

**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 the INQUIRE SYSTEM command on page 327). CVDA values are:

- **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 CICS business transaction services process of which this task is a part.

**PROCESSSTYPE**(data-area)
returns the 8-character identifier of the type definition of the CICS business transaction services 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
INQUIRE TASK

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 the INQUIRE SYSTEM command on page “INQUIRE SYSTEM options, SDTRAN” on page 325 and the CICS Resource Definition Guide 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 time-out interval, in seconds. CICS abends a task if it waits for input longer than its read time-out 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.

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 (see the CICS Front End Programming Interface User’s Guide for further information).

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 IIOP request processor is run locally the startcode for an
ASSIGN command or an INQUIRE TASK is U. When the IIOP request
processor is run remotely, over an MRO link, the startcode for these
commands is TO. (If you attempt to run the IIOP 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.

**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 a TRANSACTION resource
definition in the CICS Resource Definition Guide for more information.

CVDA values are:

**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(cva)**
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 16MB line. CVDA values are:

**ANY**
Task-lifetime storage can be either below or above the 16MB line.

**BELOW**
Task-lifetime storage must be below the 16MB line.

**TCB(cva)**
returns a CVDA value indicating the type of TCB under which the task is running. The CVDA values are:

**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 master JVM (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(cva)**
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.
INQUIRE TASK

STANTRACE

Tracing for this task is standard.

For further information on the types of tracing, see the CICS Problem Determination Guide and the description of the CETR transaction in the 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 DFHTCLnn, 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.

Description

The INQUIRE TASK LIST command returns a list of user tasks. User tasks are those 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 task executing (the one issuing the command) should be included in the task list. It is 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.

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
INQUIRE TASK LIST

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.

**Description**

The INQUIRE TCLASS command allows you to determine the current and maximum numbers of tasks within 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 command, described on page "INQUIRE TRANCLASS" on page 386, 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.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INQUIRE TCPIP returns information about the state of CICS internal sockets support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACTSOCKETS</strong>(data-value)</td>
</tr>
<tr>
<td>Returns a fullword binary field containing the current number of active IP sockets managed by the CICS sockets domain.</td>
</tr>
</tbody>
</table>

| **CRLPROFILE**(data-value) |
| Returns the name of the profile that authorizes CICS to access certificate revocation lists on the LDAP server 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. |

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
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 TCPIPSERVICE

Retrieve information about the state of a service using CICS internal TCPIP support.

Options

ATTACHSEC(cvda)
indicates, for ECI over TCP/IP services, the level of attach-time security used by connections to CICS Clients. CVDA values are:

LOCAL
CICS does not require a userid or password from clients.

VERIFY
Incoming attach requests must specify a user identifier and a user password.

This option has no meaning for Web Interface or IIOP TCP/IP services.
AUTHENTICATE returns a CVDA indicating the scheme used to authenticate clients. Possible values are:

- 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, then 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's certificate is not associated with a user ID, then HTTP basic authentication is used to obtain the client's 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's certificate is not associated with a user ID, then HTTP basic authentication is used to obtain the client's 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 IIOP.

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, then that user ID identifies the client.

This value is returned only when PROTOCOL has a value of HTTP or IIOP.

For more information about authentication and identification of HTTP and IIOP clients, see [the CICS RACF Security Guide](#).

BACKLOG returns, in fullword binary form, the maximum number of requests which can be queued in TCP/IP waiting to be processed by the service.
INQUIRE TCPIPSERVICE

CERTIFICATE(data-area)
returns a 32-character area containing the label of the certificate, within the key
ring, that is used as the server certificate in the SSL handshake for all secure
socket layer connections on this service.

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 can be between 0 and
86400 (24 hours). For the HTTP protocol, 0 should not be specified, because
this setting means that persistent connections cannot be maintained.

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
ENCYPRTION system initialization parameter, but you can edit the list to
remove or change the order of cipher suites. See the CICS RACF Security
Guide.

CONNECTIONS
returns, in fullword binary form, the number of sockets connections for this
service.

DNSGROUP(data-area)
returns the 18–character DNS group name that this TCPIPSERVICE registers
with the OS/390 Workload Manager (WLM).

DNSSTATUS(cvda)
returns the current state of WLM/DNS registration of this TCPIPSERVICE. The
CVDA values are:

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 OS/390.

UNREGISTERED
Registration has not yet occurred (this 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.

GRPCCRITICAL(cvda)
returns a CVDA value specifying whether or not this TCPIPSERVICE is a
critical member of the DNS group. The CVDA values are:

CRITICAL
If this TCPIPSERVICE 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 is closed, or abnormally stops listening for any
INQUIRE TCPIPSERVICE

reason, the group name specified in the DNSGROUP attribute is not deregistered from WLM, unless this is the last service in a set with the same group name.

**IPADDRESS** *(data-area)*
returns the 15-character dotted decimal IP address of this service.

**MAXDATALEN** *(data-area)*
returns, in fullword binary form, the setting for the maximum length of data that may be received by CICS as an HTTP server.

**NUMCIPHER** *(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 value indicating the status of the CICS Web Interface. CVDA values are:

- **OPEN**
  CICS internal sockets support is open for this service.

- **CLOSED**
  CICS internal sockets support has not yet been activated, or has been terminated, for this service.

- **CLOSING**
  CICS internal sockets support is in the process of closing for this service.

- **IMMCLOSING**
  CICS internal sockets support is in the process of immediate termination.

**PORT**
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:

- **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 value indicating the underlying protocol being used on this service. CVDA values are:

- **ECI**
  External CICS interface protocol.

- **HTTP**
  Hypertext Transfer protocol.

- **IIOP**
  Internet Inter-orb protocol.

- **USER**
  User-defined protocol.
INQUIRE TCPIPSERVICE

SOCKETCLOSE(cvda)
returns whether a TIMEOUT value is in effect for this service. CVDA values are:

WAIT  NO was specified on the definition. Socket receives will 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 value specifying the level of secure sockets support being used for this service. CVDA values are:

NOSSL  Secure Sockets Layer is not being used for this service.

SSL  Secure Sockets Layer without client authentication is being used for this service.

CLIENTAUTH  Secure Sockets Layer with 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)
returns the 6-character prefix of the temporary storage queue used to store inbound data and Web documents created by applications. The TS queue prefix must be matched by a corresponding TSMODEL definition to meet your system and application requirements.

URM(data-area)
returns the 8-character name of the service user-replaceable module invoked by 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  TCPIP not available (TCPIP=NO was specified as a system initialisation parameter)

5  TCPIP is closed.

NOTAUTH
RESP2 values:
INQUIRE TCPIPSERVICE

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

NOTFND
  RESP2 values:
  3  The TCPIPSERVICE was not found
INQUIRE TDQUEUE

Retrieves information about a transient data queue.

### 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.

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
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 may 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 (REOTENAME), and 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 21 for general information about browsing, including syntax, exception conditions, and examples.

**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:

- **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 executed 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 associated with the queue. If the queue is not intrapartition, or no transaction is defined for it (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:

- **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 through 32767).

DATABUFFERS(data-area) (extrapartition queues only)
returns the number of buffers (in the range 1 through 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 may refer to a data set name used in the startup JCL.

DISPOSITION(cvda) (extrapartition queues only)
returns a CVDA value indicating the status of the associated data set. CVDA values are:

- **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 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:

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:

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 should take if an I/O error is encountered. CVDA values are:

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 that this indirect queue points to. 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 CICS is to take for an in-doubt unit of work (UOW) if the definition for this queue specifies WAIT(YES). CVDA values are:

**QUEUE**
The UOW is in-doubt and waiting; any locks held by the UOW for this queue remain active until the final state of the UOW is known. This means that 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 in-doubt UOW had issued READQ or DELETEQ requests.
- WRITEQ if the in-doubt UOW had issued WRITEQ or DELETEQ requests.
- DELETEQ if the in-doubt UOW had issued READQ, WRITEQ, or DELETEQ requests.

**REJECT**
The UOW is in-doubt 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 in-doubt unit of work (UOW), which has modified a recoverable queue, should wait for resynchronization with its coordinator to determine whether to commit or backout the changes. CVDA values are:

**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 UOW's transaction definition. See the CICS Resource Definition Guide for an explanation of the interactions of in-doubt attributes on the TDQUEUE and TRANSACTION definitions.

IOTYPE(cvda) (extrapartition queues only)
returns a CVDA value indicating whether the queue was defined for INPUT, OUTPUT, or RDBACK. CVDA values are:

**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.
INQUIRE TDQUEUE

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:

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 the 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:

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:

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
INQUIRE TDQUEUE

having fixed-length records, or the maximum record length for queues having variable-length records. RECORDLENGTH 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:

LOGICAL
The queue is logically recoverable.

NOTAPPLIC
The queue is not intrapartition.

NOTRECOVABLE
The queue is not recoverable.

PHYSICAL
The queue is physically recoverable.

REMTENAME(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). REMOTENAME applies only to queues defined as remote; for other queues the value returned is blanks.

REMTESYSTEM(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). REMOTESYSTEM 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:

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 DNAME 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 the queue must reach before automatic transaction initiation (ATI) occurs. When the queue reaches this depth, CICS invokes a task to process it automatically. A value of zero means 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:

EXTRA
The queue is extrapartition.

INDIRECT
The queue is indirect.
INQUIRE TDQUEUE

**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 TERMINAL

INQUIRE TERMINAL

Retrieve information about a terminal or session.

INQUIRE TERMINAL/NETNAME

INQUIRE TERMINAL(data-value) NETNAME(data-area)

The following options apply to both the INQUIRE TERMINAL and the INQUIRE NETNAME command.
For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.
INQUIRE TERMINAL

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:

• 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 which 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.

You can also use INQUIRE TERMINAL to inquire about an APPC, LUTYPE6.1, or MRO session or, where there are parallel sessions, session group. To get full details about the associated connection, however, you must use an INQUIRE CONNECTION command.

Some of the options in this command return system status information, such as whether the terminal is acquired or not, whether it is in use by a task, and so on. 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 may have been autoinstalled. For a session, the CONNECTION defines shared properties and SESSIONS defines specifics. See the CICS Resource Definition Guide for more information about TERMINAL, TYPETERM, SESSIONS, and CONNECTION resource definitions.

In most cases, options of this type have the same name—or one recognizably similar—as 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 appear 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.

**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 is called the terminal-owning region (TOR). Note that
different terminals may 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 for the link. (It can name an indirect
connection, but that is an unusual setup). In this case, the netname of the TOR is
specified in the link CONNECTION definition.

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, as follows:

- 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 a fuller explanation of the
relationship between REMOTESYSTEM, REMOTESYSNET, and LINKSYSTEM,
see “Remote connections” on page 184.

**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”
on page 21 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.
CVDA values are:

- **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.
INQUIRE TERMINAL

NOTAPPLIC
The terminal is an MRO session.

TCAM  The access method is TCAM (DCB interface).

VTAM  The access method is VTAM.

ACQSTATUS(cvda)  (VTAM only)
returns the same value as the TERMSTATUS option and is retained only for compatibility purposes. You should use TERMSTATUS in new applications.

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. CVDA values are:

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 VTAM 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 DEFSRNHT and SCRNHT options.

ALTSCRNWD(data-area)
returns a halfword binary field giving the width (in characters) of the alternate screen size. See also the DEFSRNWD 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 is a character field and, in that context, null refers to the blank character x'40', see [Null values on page 14] for a discussion of null values.

APLKYBDST(cvda)
returns a CVDA value indicating whether the terminal has the APL keyboard feature. CVDA values are:

APLKYBD
The terminal has the APL keyboard feature.
INQUIRE TERMINAL

**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. CVDA values are:

**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. CVDA values are:

**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. CVDA values are:

**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 should attempt to establish (bind) a session with this terminal when communication with VTAM is established. CVDA values are:

**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.
INQUIRE TERMINAL

NOTAPPLIC
The terminal is not a VTAM 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. CVDA values are:

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. CVDA values are:

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. CVDA values are:

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 only set for sessions, as follows:
• 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) (VTAM only)
returns a CVDA value indicating whether CICS should attempt to acquire the terminal if it is required for an automatic task initiation (ATI) request. Only VTAM physical terminals can be acquired by CICS; sessions are not eligible. CVDA values are:
INQUIRE TERMINAL

CREATE
The terminal can be acquired.

NOCREATE
The terminal cannot be acquired.

NOTAPPLIC
The terminal is not a VTAM 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. CVDA values are:

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" on page 668.

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. CVDA values are:

DISCREQ
CICS will honor a request to disconnect this terminal (with a VTAM CLSDST request to terminate the session if the terminal is a VTAM terminal).

NODISCREQ
CICS will not honor a request to disconnect this terminal.

DUALCASEST(cvda)
returns a CVDA value indicating whether the terminal has a typewriter keyboard or an operator console keyboard. CVDA values are:

DUALCASE
The terminal has a typewriter keyboard.
INQUIRE TERMINAL

NODUALCASE
The terminal has an operator console keyboard (this keyboard is not restricted to a single case), or is not a 3270 display.

EXITTRACING(*cvda) (VTAM only)
returns a CVDA value indicating whether this terminal is traced when CICS VTAM exit tracing is active. (See the TCXEXITSTATUS option in the INQUIRE TRACEFLAG command, on page "TCEXITSTATUS" on page 382.) CVDA values are:

EXITTRACE
The terminal is traced.

NOEXITTRACE
The terminal will not be traced.

NOTAPPLIC
The terminal is not a VTAM terminal, or is a remote terminal, a surrogate terminal, or a model definition.

EXTENDEDSS(*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 EXTENDEDSS option) or implicitly, by specifying features that use the extended data stream (see the BACKTRANS, COLORST, HILIGHTST, 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. CVDA values are:

EXTENDEDSS
The terminal supports the extended data stream.

NOEXTENDEDSS
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. CVDA values are:

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. CVDA values are:

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
INQUIRE TERMINAL

(GCSGID), which identifies the set of graphic characters that can be 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. CVDA values are:

HFORM
The terminal has the horizontal forms feature.

NOHFORM
The device does not have the horizontal forms feature.

HILIGHTST(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. CVDA values are:

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. CVDA values are:

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. CVDA values are:

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
INQUIRE TERMINAL

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 may 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 may 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, may 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 may
be suffixed by a terminal or alternate suffix. If this terminal is a surrogate, the
mapset name may 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.

Note: See the CICS Application Programming Guide for information about
mapset suffixing.

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.

MSRCONTROL(cvda)
returns a CVDA value indicating whether the terminal has a magnetic slot
reader. This feature is available only on 8775 and 3643 terminals. CVDA values
are:

MSRCONTROL
The terminal has a magnetic slot reader.

NOMSRCRCONTROL
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 in the CICS Application Programming Reference manual. 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. CVDA
values are:

MODEL
A remote terminal definition (representing a terminal owned by another
CICS region) which is not currently expanded into a surrogate.
INQUIRE TERMINAL

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 is the name by which this terminal is known to VTAM. For ISC sessions, it is the name by which the session (or session group, if there are parallel sessions) is known to VTAM. 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 VTAM in the remote region. (For a remote terminal routed from a pre-CICS Transaction Server for z/OS region, NETNAME is blank.)

If the netname is a VTAM LU alias, it is different from the netname component of the NQNAME, which always contains the real netname.

**Note:** 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 on page "INQUIRE NETNAME" on page 278.

**NEXTTRANSID**(data-area)

returns the 4-character identifier of the transaction to be executed 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 blanks 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 VTAM at logon time.

This 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 VTAM. 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-VTAM or a remote terminal, NQNAME is blank. If the resource is a VTAM resource but has not yet received an NQNAME, CICS returns the known netname.
**INQUIRE TERMINAL**

**OBFORMATST(cvda)**  
returns a CVDA value indicating whether outboard formatting can be used for this terminal. CVDA values are:

**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 only applies to the 3790 and 3770 batch data interchange logical units. CVDA values are:

**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.

**Note:** If the terminal is a surrogate terminal, this value may 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, who may since have signed off. For information, see the [CICS RACF Security Guide](../). The OPERID may 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. CVDA values are:

**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.

**PAGESIZESTATUS(cvda)**  
returns a CVDA value indicating how pages of BMS messages with a disposition of PAGING should be delivered to the terminal. CVDA values are:

**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. CVDA values are:

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. CVDA values are:

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. CVDA values are:

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. CVDA values are:

NOPRTCOPY
CICS is not to use the hardware copy feature.

NOTAPPLIC
The terminal is not a VTAM 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 should use a QUERY structured field to determine the characteristics of the terminal.

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.
INQUIRE TERMINAL

RELREQST\texttt{(cvda)} (VTAM only) returns a CVDA value indicating whether CICS is to honor requests from VTAM to release the terminal or session. CVDA values are:

\begin{itemize}
\item \textbf{NORELREQ} CICS cannot release the logical unit, or the access method is not VTAM.
\item \textbf{RELREQ} CICS can release the logical unit.
\end{itemize}

\begin{itemize}
\item \textbf{REMOTE\textsc{NAME}(data-area)} returns the 4-character name of this terminal in the remote CICS region in which it is defined. \textsc{REMOTE\textsc{NAME}} applies only to terminals defined as remote; for others the value returned is blanks.
\item \textbf{REMOTES\textsc{YSNET}(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 \textsc{REMOTESYSTEM} field has not been installed, and no value was specified for the \textsc{REMOTESYSNET} option when the terminal was defined.
\item \textbf{REMOTES\textsc{YSTEM}(data-area)} returns the 4-character name 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.
\end{itemize}

Otherwise this field is blank.

\begin{itemize}
\item \textbf{SCR\textsc{HNHT}(data-area)} (or \textsc{SCREEN\textsc{HEIGHT}}) returns a halfword binary field giving the height (in lines) of the current screen size. See also the \textsc{DEFSCR\textsc{HNHT}} and \textsc{ALTSCR\textsc{HNHT}} options.
\end{itemize}

\begin{itemize}
\item \textbf{Note:} \textsc{SCR\textsc{HNHT}} is a synonym for the \textsc{SCREEN\textsc{HEIGHT}} option of earlier releases of CICS. For compatibility, CICS recognizes \textsc{SCREEN\textsc{HEIGHT}} as equivalent.
\end{itemize}

\begin{itemize}
\item \textbf{SCR\textsc{NW\textsc{D}(data-area)} (or \textsc{SCREEN\textsc{WIDTH}}) returns a halfword binary field giving the current width of the terminal screen (in characters). See the \textsc{DEFSCR\textsc{NW\textsc{D}} and \textsc{ALTSCR\textsc{NW\textsc{D}} options.}
\end{itemize}

\begin{itemize}
\item \textbf{Note:} \textsc{SCR\textsc{NW\textsc{D}} is a synonym for the \textsc{SCREEN\textsc{WIDTH}} option of earlier releases of CICS. For compatibility, CICS recognizes \textsc{SCREEN\textsc{WIDTH}} as equivalent.
\end{itemize}

\begin{itemize}
\item \textbf{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. CVDA values are:
\begin{itemize}
\item \textbf{NOPRESETSEC} The terminal does not have preset security.
\item \textbf{PRESETSEC} The terminal has preset security.
\end{itemize}
\end{itemize}

\begin{itemize}
\item \textbf{SERV\textsc{STATUS(cvda)}} returns a CVDA value indicating whether the terminal is available for use (from the point of view of the local CICS system, which may be different from the system which owns the terminal). \textsc{SERV\textsc{STATUS}} corresponds to the
INQUIRE TERMINAL

INSERVICE option in the TERMINAL definition. “Available” (INSERVICE) does not necessarily imply, for a VTAM terminal, that the terminal is acquired.

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 VTAM sessions. CVDA values are:

APPCPARALLEL
This is a parallel APPC session group.

APPCSINGLE
This is a single APPC session.

LU61 This is 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. CVDA values are:

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. CVDA values are:

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 executing at this terminal. Zero is returned if no task is using the terminal.

TCAMCONTROL(data-area) (TCAM only)
returns a 1-character TCAM control byte giving one of the following codes to identify which segment of a message has passed between CICS and TCAM.

The meanings are:

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
**INQUIRE TERMINAL**

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) **(VTAM only)**  
returns a CVDA value indicating whether CICS is in session with the logical unit  
represented by this terminal. CVDA values are:

- **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 VTAM 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. CVDA values are:

- **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. CVDA values are:

- **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.  
CVDA values are:

- **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

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.

TTSTATUS

returns a CVDA value indicating whether this terminal can initiate tasks by entering unsolicited input. CVDA values are:

NOTTI This terminal cannot initiate transactions.

TTI This terminal can initiate transactions.

UCTRANST

returns a CVDA value indicating whether input from this terminal is translated to uppercase 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; there is also a UCTRAN option in a PROFILE definition, but that value is not pertinent here. CVDA values are:

NOUCTRAN Input from this terminal is not translated to uppercase on receipt. (It is translated before presentation to the task issuing a RECEIVE, however, if the PROFILE definition for the transaction being executed specifies translation. See Table 6 on page 600 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 execute, that decision is made from a copy of the input which has been translated to uppercase. There is no difference in the data presented to the task between these two options.

UCTRAN The input is translated to uppercase on receipt. (It is unaffected by the translation option in the PROFILE).

USERAREA

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'.
INQUIRE TERMINAL

USERID(data-area)
returns the 8-character identifier of the user signed on at this terminal or session.

If there is no signed-on user, 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 there is no signed on user.

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, additional to normal 3270 function. This feature is available only on 8775 and 3290 terminals. CVDA values are:

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. CVDA values are:

NOVFORM
The device does not have the vertical forms feature.

VFORM
The terminal has the vertical forms feature.

ZCPTRACING(cvda) (VTAM only)
returns a CVDA value indicating whether this terminal is traced when CICS tracing for VTAM terminals is turned on. CVDA values are:

NOTAPPLIC
The terminal is not a VTAM terminal, or is a surrogate terminal or a model definition.

NOZCPTRACE
The terminal will not be traced.

ZCPTRACE
The terminal is traced.

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.
INQUIRE TERMINAL

NOTAUTH
  RESP2 values:
  100 The user associated with the issuing task is not authorized to use this command.

TERMIDERR
  RESP2 values:
  1 The named terminal cannot be found.
INQUIRE TRACEDEST

Retrieve information about tracing.

### 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.

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
INQUIRE TRACEDEST

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.

### 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
INQUIRE TRACEFLAG

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) (VTAM only)
returns a CVDA value indicating which invocations of the CICS VTAM exits are being traced.

Two types of exit activity can be traced: invocations associated with particular terminals that have been designated for VTAM exit tracing (“terminal-specific” activity) and invocations not related to any particular terminal (“nonterminal-specific” activity).

CVDA values are:

**NOTAPPLIC**
VTAM 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.

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 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
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BA</td>
<td>Business applications manager</td>
</tr>
<tr>
<td>BF*</td>
<td>Built-in functions</td>
</tr>
<tr>
<td>BM*</td>
<td>Basic mapping support</td>
</tr>
<tr>
<td>BR*</td>
<td>BRIDGE 3270 Bridge</td>
</tr>
<tr>
<td>CP*</td>
<td>CPI Common programming interface</td>
</tr>
<tr>
<td>DC*</td>
<td>Dump control</td>
</tr>
<tr>
<td>DD</td>
<td>DIRMGR Directory manager</td>
</tr>
<tr>
<td>DH</td>
<td>Document handling</td>
</tr>
<tr>
<td>DI*</td>
<td>Batch data interchange</td>
</tr>
<tr>
<td>DM</td>
<td>DOMAINDMGR Domain manager</td>
</tr>
<tr>
<td>DP</td>
<td>Debugging Profiles domain</td>
</tr>
<tr>
<td>DS</td>
<td>DISPATCHER Dispatch manager</td>
</tr>
<tr>
<td>DU</td>
<td>DUMP Dump manager</td>
</tr>
<tr>
<td>EI*</td>
<td>EXEC interface</td>
</tr>
<tr>
<td>EJ</td>
<td>ENTJAVA Enterprise Java domain</td>
</tr>
<tr>
<td>EM</td>
<td>EVENTMGR Event manager</td>
</tr>
<tr>
<td>FC*</td>
<td>File control and DL/I</td>
</tr>
<tr>
<td>GC</td>
<td>GLOBALCATLG CICS global catalog manager</td>
</tr>
<tr>
<td>IC*</td>
<td>Interval control</td>
</tr>
<tr>
<td>IE</td>
<td>ECI over TCP/IP domain</td>
</tr>
<tr>
<td>II</td>
<td>IIOP domain</td>
</tr>
<tr>
<td>IS*</td>
<td>Intersystem communication</td>
</tr>
<tr>
<td>KC*</td>
<td>Task control</td>
</tr>
<tr>
<td>KE</td>
<td>KERNEL Kernel</td>
</tr>
<tr>
<td>LC</td>
<td>LOCALCATLG CICS local catalog manager</td>
</tr>
<tr>
<td>LD</td>
<td>LOADER Program load manager</td>
</tr>
<tr>
<td>LG</td>
<td>LOGMGR Log manager</td>
</tr>
<tr>
<td>LM</td>
<td>LOCKMGR Lock manager</td>
</tr>
<tr>
<td>ME</td>
<td>MESSAGE Message manager</td>
</tr>
<tr>
<td>MN</td>
<td>MONITOR Monitoring manager</td>
</tr>
<tr>
<td>NQ</td>
<td>Enqueue domain</td>
</tr>
<tr>
<td>OT</td>
<td>OBJECTTRAN Object Transaction Service (OTS) domain</td>
</tr>
<tr>
<td>PA</td>
<td>PARAMGR Parameter manager</td>
</tr>
<tr>
<td>PC*</td>
<td>Program control</td>
</tr>
<tr>
<td>PG</td>
<td>PROGMGR Program manager</td>
</tr>
<tr>
<td>PI</td>
<td>Pipeline manager</td>
</tr>
<tr>
<td>PT</td>
<td>Partner manager</td>
</tr>
<tr>
<td>RI*</td>
<td>Resource manager interface (RMI)</td>
</tr>
<tr>
<td>RM</td>
<td>Recovery manager</td>
</tr>
<tr>
<td>RX</td>
<td>RRS Resource recovery services</td>
</tr>
<tr>
<td>RZ</td>
<td>REQUESTSTRM Request streams domain</td>
</tr>
<tr>
<td>SC*</td>
<td>Storage control</td>
</tr>
<tr>
<td>SH</td>
<td>Scheduler services domain for BTS</td>
</tr>
<tr>
<td>SJ</td>
<td>SJVM CICS JVM domain</td>
</tr>
<tr>
<td>SM</td>
<td>STORAGE Storage manager</td>
</tr>
<tr>
<td>SO</td>
<td>SOCKETS Sockets</td>
</tr>
<tr>
<td>ST</td>
<td>STATISTICS Statistics manager</td>
</tr>
<tr>
<td>SZ*</td>
<td>Front-end programming interface</td>
</tr>
<tr>
<td>TC*</td>
<td>Terminal control</td>
</tr>
<tr>
<td>TD*</td>
<td>Transient data</td>
</tr>
<tr>
<td>TI</td>
<td>TIMER Timer manager</td>
</tr>
<tr>
<td>TR</td>
<td>TRACE Trace manager</td>
</tr>
<tr>
<td>TS</td>
<td>Temporary storage</td>
</tr>
<tr>
<td>UE*</td>
<td>User exit interface</td>
</tr>
</tbody>
</table>
INQUIRE TRACETYPE

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>US</td>
<td>USER</td>
<td>User interface</td>
</tr>
<tr>
<td>WB</td>
<td>WEB</td>
<td>Web interface</td>
</tr>
<tr>
<td>XM</td>
<td>TRANMGR</td>
<td>Transaction manager</td>
</tr>
<tr>
<td>XS</td>
<td>SECURITY</td>
<td>Security manager</td>
</tr>
</tbody>
</table>

SPECIAL

indicates that CICS should return the trace levels for special tracing.

STANDARD

indicates that CICS should return the trace levels for standard tracing.

Conditions

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

The INQUIRE TRANCLASS command allows you 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 wish. See "Browsing resource definitions" on page 21 for general information about browsing, including syntax, exception conditions, and examples.

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.

MAXACTIVE(data-area)
returns a fullword binary field giving the largest number of tasks in the transaction class which 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 DFHTCLnn, where nn is the two-digit class number.

Conditions: INVREQ, NOTAUTH, TCIDERR
INQUIRE TRANCLASS

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

INQUIRE TRANDUMPCODE

Retrieve information about a transaction dump code.

INQUIRE TRANDUMPCODE

INQUIRE TRANDUMPCODE(data-value)

- CURRENT(data-area)
- DUMPSCOPE(cvda)
- MAXIMUM(data-area)
- SHUTOPTION(cvda)
- SYSDUMPING(cvda)
- TRANDUMPING(cvda)

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 TRANSDUMPCODE 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.
**INQUIRE TRANDUMPCODE**

**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 21 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-specified, 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.

**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
INQUIRE TRANDUMPCODE

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:

NOSYSDUMP
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.

**Description**

The INQUIRE TRANSACTION command retrieves information about a particular transaction installed in your CICS system.

**Conditions:** END, ILLOGIC, INVREQ, NOTAUTH, TRANSIDERR

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
INQUIRE TRANSACTION

Most of the values come from the TRANSACTION resource definition, but a few come from the profile definition to which it points (these are noted in the descriptions). See the CICS Resource Definition Guide 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 executing 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 may be different from the one executed 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 wish. See “Browsing resource definitions” on page 21 for general information about browsing, including syntax, exception conditions, and examples.

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.

CMDSEC(cvda)
returns a CVDA value indicating whether command security checking should be performed for tasks executing this transaction. CVDA values are:

CMDSECNO
Command security checking should not be performed.

CMDSECYES
Command security checking should be performed.

DTIMEOUT(data-area)
returns a fullword binary field giving the deadlock time-out value (in seconds) for a task executing this transaction. CICS abends a task that waits for a locked resource longer than its deadlock timeout value.

DUMPING(cvda)
returns a CVDA value indicating 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.
INQUIRE TRANSACTION

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 in-doubt period.

The action depends on the values returned in INDOUBTWAIT and INDOUBTMINS; if INDOUBTWAIT returns WAIT, the action is not normally taken until the time returned in INDOUBTMINS expires. (For exceptions to this rule, see INDOUBTWAIT.)

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 in-doubt 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 in-doubt 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 in-doubt state. CVDA values are:
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.

Note: Even if INDOUBTWAIT returns WAIT, there may be aspects of the UOW that force CICS to take an immediate decision—that is, to take immediately the action specified on the ACTION attribute of the transaction definition. This can happen if, for example, the UOW contains:
  - Subordinate LU6.1 sessions
  - Subordinate MRO sessions to pre-CICS Transaction Server for z/OS systems.
INQUIRE TRANSACTION

For further information about the meaning of the ACTION and WAIT attributes of the TRANSACTION definition, see the CICS Resource Definition Guide.

ISOLATEST(cvda)

returns a CVDA value indicating whether a task executing this transaction should 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 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 non-task storage.

Isolation must be turned on for the system as well as the transaction in order for a task to run isolated. (See the TRANISOLATE option of the INQUIRE SYSTEM command.) CVDA values are:

**ISOLATE**
Tasks should run isolated.

**NOISOLATE**
Tasks should not run isolated.

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 prior to syncpoint.

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 executing the transaction and the terminal or session which is its principal facility.

PROGRAM(data-area)

returns the 8-character name of the first program invoked by a task executing this transaction.

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.

REMTENANE(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. (Read the description of
INQUIRE TRANSACTION

"Defining a TRANSACTION" in the CICS Resource Definition Guide (for a fuller discussion of the length of REMOTENAME). Blanks are returned if the transaction is not remote.

**REMOTESYSTEM**(data-area)
returns the 4-character name 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 remotesystem 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 should be performed for a task executing this transaction. CVDA values are:

RESSECNO
Resource-level checking should not be performed.

RESSECYES
Resource-level checking should be performed.

**ROUTING**(cvda)
returns a CVDA value indicating whether a task executing this transaction is subject to dynamic routing. CVDA values are:

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:

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 will be 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 executing this transaction, in seconds. CICS abends 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 executing this transaction. 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.
INQUIRE TRANSACTION

RUNAWAYTYPE(cvda)
returns a CVDA value indicating the source of the RUNAWAY option value for this transaction. CVDA values are:

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.

SCRSIZE(cvda)
returns a CVDA value indicating whether a task executing this transaction should use the alternate or the default screen size. This value is defined in the profile definition (see the PROFILE option). CVDA values are:

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 executed during CICS shutdown by a task created to process unsolicited 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)
returns a CVDA value indicating whether the transaction is available for use. CVDA values are:

DISABLED
   The transaction is not available for use.

ENABLED
   The transaction is available for use.

STORAGECLEAR(cvda)
returns a CVDA value indicating whether CICS should clear storage that is released from a task executing this transaction (to prevent other tasks accidentally viewing confidential data). CVDA values are:

CLEAR
   Storage is cleared.

NOCLEAR
   Storage will not be cleared.

TASKDATAKEY(cvda)
returns a CVDA value indicating the key of the storage CICS assigns to a task executing this transaction. This storage includes 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 the task.

CVDA values are:

CICSDATAKEY
   CICS-key storage is assigned.
INQUIRE TRANSACTION

USERDATAKEY
User-key storage is assigned.

TASKDATALOC(cvda)
returns a CVDA value indicating whether task-lifetime storage for a task executing this transaction should be above or below the 16MB line. Task-lifetime storage includes the EIB and TWA. CVDA values are:

ANY Task-lifetime storage can be above or below the 16MB line.

BELOW Task-lifetime storage must be below the 16MB 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 DFHTCLnn, where nn is a number from 00 to 10, and it is this number that 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 executing this transaction. CVDA values are:

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 executing. 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.
INQUIRE TRANSACTION

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 could not be found.
INQUIRE TSMODEL

Retrieve information about a temporary storage model.

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.

Description

The INQUIRE TSMODEL command returns information about a particular TS model.

Browsing

You can also browse through all of the temporary storage models in your system by using the browse options (START, NEXT, and END) on INQUIRE TSMODEL commands.

See "Browsing resource definitions" on page 21 for general information about browsing, including syntax, exception conditions, and examples.

Options

LOCATION(cvda)
returns a CVDA value indicating where queues matching the model are to be stored. CVDA values are:

- 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 hex 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:
INQUIRE TSMODEL

RECOVERABLE
Queue names matching this model are recoverable.

NOTRECOVABLE
Queue names matching this model are non-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:

SECURITY
Security checking is performed for queue names matching this model.

NOSECURITY
Security checking will not be 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.

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 TSMODEL does not exist.
INQUIRE TSPOOL

Retrieve information about a shared temporary storage pool.

```
INQUIRE -TSPOOL(data-value)

CONNSTATUS(cvda)
```

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 TSPOOL command returns information about a particular shared temporary storage pool.

Browsing

You can also browse through all of the temporary storage pools in your system by using the browse options (START, NEXT, and END) on INQUIRE TSPOOL commands.

See "Browsing resource definitions" on page 21 for general information about browsing, including syntax, exception conditions, and examples.

Options

```
TSP0OL(data-value)
```
returns an 8-character field giving the shared TS pool name.

```
CONNSTATUS(cvda)
```
returns a CVDA value containing the connection status of this pool. CVDA values are:

CONNECTED
This pool is connected.

UNCONNECTED
This pool is not connected.

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:
INQUIRE TSPOOL

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 section applies also to the alternative command, INQUIRE TSQNAME. Use either to inquire about names up to 8 characters long, use INQUIRE TSQNAME to inquire about names up to 16 characters long.

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.

Description

The INQUIRE TSQUEUE command returns information about a particular temporary storage queue.

Browsing

You can also browse through all of 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 wish. If you want to see all the queues with names beginning with a certain string of letters, for example, you can start your browse with an AT value of those letters, padded on the right to eight characters with nulls (X'00'). If you want to browse TS queues that are in a shared temporary storage pool managed by a TS server, you must specify the POOLNAME or the SYSID option on the browse START request only. If CICS cannot find the specified SYSID in any temporary storage table (TST) TYPE=SHARED entry, CICS returns the INVREQ condition.

Note: If you do a WRITEQ, for example, to queue, which maps to a shared TS pool because of a TST definition, be aware that to inquire on this queue you need to specify the explicit SYSID on the INQUIRE command.

In a browse, CICS returns all queues, and you may see queues created by CICS for internal use as well as those created by user applications. In particular, queues with names that start with these characters are CICS queues: ‘**’, ‘$$’, ‘X’FF’, ‘CEBR’ and ‘DF’.
See “Browsing resource definitions” on page 21 for general information about browsing, including syntax, exception conditions, and examples.

Options

**FLENGTH**(*data-area*)
returns a fullword binary field giving 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 has been stored in a single entry in the coupling facility (in which case the returned value for FLENGTH is less than 32K (32768)), this is the total size of all items including their control information.

When the shared queue has been stored as a separate list in the coupling facility, the total size is estimated as MAXITEMLEN times NUMITEMS.

**LASTUSEDINT**(*data-area*)
returns a fullword binary field specifying 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 the **CICS System Definition Guide** for further information).

**LOCATION**(*cvda*)
returns a CVDA value indicating where the temporary storage queue resides. CVDA values are:

**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 giving 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 storage queues, 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 (depending on the control interval size of the temporary storage data set). (For background information about CI sizes, see the **CICS System Definition Guide**)

For shared queues, the lengths returned in MINITEMLEN, MAXITEMLEN, and FLENGTH, reflect the data length stored in the coupling facility. This 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 will be capped at 32767 (Hex ’7FFF’).

**MINITEMLEN**(*data-area*)
returns a halfword binary field giving the length in bytes of the smallest item in the temporary storage queue. For information about how CICS calculates the length of items, and for information about shared TS queues, see the **MAXITEMLEN** option.
INQUIRE TSQUEUE / TSQNAME

NUMITEMS (data-area)
returns a halfword binary field giving 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:

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.

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 which 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 8 significant characters. The queue name is truncated, some significant characters are lost.
4 This temporary storage queue name cannot be deleted as it was written by CICS using the PUTQ macro.
INQUIRE TSQUEUE / TSQNAME

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).

**Diagram:**

```
INQUIRE UOW(data-area)
  AGE(data-area)
  LINK(data-area)
  NETNAME(data-area)
  NETUOWID(data-area)
  OTSTID(data-area)
  SYSID(data-area)
  TASKID(data-area)
  TERMID(data-area)
  TRANSID(data-area)
  UOWSTATE(cvda)
  USERID(data-area)
  WAITCAUSE(cvda)
  WAITSTATE(cvda)
```

**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.

Chapter 2. System commands 407
INQUIRE UOW

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 21 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.

Note: 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.

For examples of the use of the INQUIRE UOW command, see the CICS Problem Determination Guide.

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 (including OTSCOORD), 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 (including OTSCOORD), 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, CSMI, 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 in-doubt 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 in-doubt 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 in-doubt 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.

**OTSCOORD**
This UOW is waiting or has been shunted because of the failure of a
session to the OTS coordinator. If OTSCOORD is returned, the AGE,
HOST, NETUOWID, OTSTID, TASKID, TRANSID, UOWSTATE,
USERID, and WAITSTATE attributes are meaningful.

**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 syncpoint.
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**
Syncpoint 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**
Syncpoint 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 UOWs that have updated CICS file control-managed data sets.

**Browse only function**

The INQUIRE UOWDSNFAIL command can be used only in browse mode.

**INQUIRE UOWDSNFAIL**

```
INQUIRE UOWDSNFAIL
  CAUSE(cvda)
  DSNAME(data-area)
  NETNAME(data-area)
  REASON(cvda)
  RLSACCESS(cvda)
  SYSID(data-area)
  UOW(data-area)
```

**Conditions:** END, ILLOGIC, NOTAUTH

**Description**

This command enables you to inquire on the reasons why UOWs were shunted because of a failure during syncpoint associated with a specified data set. Failures during syncpoint processing result in locks held by the UOW against the data set (or data sets) which suffered the failure being retained. Thus, when a failure is reported by this command, it also indicates the presence of retained locks.

The UOWDSNFAIL command returns UOWs that are shunted and also UOWs that are in the process of being retried. In the latter case, the only data sets returned are those that have not yet been processed as part of the retry.

Note that there may be failures 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 Recovery and Restart 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.

**Browsing**

You can use the browse options (START, NEXT, and END) to find all the units of work with syncpoint failures, together with the data sets that have suffered failures. In addition, the reason is given for each unique UOW/data set combination (a UOW can have syncpoint failures for several data sets but, for each data set within the UOW, the cause of the failure is the same). See "[Browsing resource definitions](#)" for general information about browsing, including syntax, exception conditions, and examples.

Because this command returns information about UOWs that are currently failed with respect to data sets (with associated retained locks held against those data sets), it does not return information about failures that are in the process of being retried when the command is issued. For example, if a UOW suffered a backout failure with respect to a particular data set, and a SET DSNAME RETRY command was issued, the INQUIRE UOWDSNFAIL command would not return information about that backout failure.
INQUIRE UOWDSNFAIL

was issued for that data set, that particular UOW/data set combination would not appear in the browse. The backout retry might either be successful, in which case the failure condition will have been cleared, or it might fail again, in which case the UOW/data set combination would appear if a new INQUIRE UOWDSNFAIL browse were started.

One important use of this command is to enable you to write a transaction that helps operators to identify and remove retained locks, so that data sets can be quiesced and used for batch application programs. There are several CICS-supplied sample programs that you can use unmodified, or use as a basis for writing your own programs. See the sample application programs, DFH0BAT1 through DFH0BAT8, for a working illustration of the use of this command. These are supplied in the CICSTS31.CICS.SDFHSAMP library, and are described in the CICS Operations and Utilities Guide.

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 datasets 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 indicating which failed component has caused the UOW to have retained locks for this data set. CVDA values are:

CACHE
A VSAM RLS cache structure, or connection to it, has failed.

CONNECTION
An intersystem connection error has caused the UOW to fail while in-doubt. The name of the system to which connectivity was lost is returned on the SYSSID 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 suffered a backout failure in this UOW.
NETNAME \textit{(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 \textit{(cvda)}
returns a CVDA value (when the CVDA returned on the CAUSE parameter is RLSSERVER, CONNECTION, or DATASET) indicating the specific reason for the error against this data set. CVDA values are:

\begin{itemize}
\item \textbf{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.
\item \textbf{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.
\item \textbf{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 DSNAM RETRY. For further information, see the section on moving data sets in the \textit{CICS Recovery and Restart Guide}.
\item \textbf{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.
\item \textbf{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.
\item \textbf{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.
\item \textbf{INDEXRECFULL} 
A larger alternate index record size needs to be defined for the data set. For further information, see the section on moving data sets in the \textit{CICS Recovery and Restart Guide}.
\item \textbf{INDEXRECFULL} 
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.
\item \textbf{INDOUBT} 
The unit of work had issued recoverable requests against the data set, and has now failed in-doubt. The connection to the coordinating system needs to be reestablished.
\item \textbf{IOERROR} 
A hard I/O error occurred during backout. To correct this error, restore a
INQUIRE UOWDSNFAIL

full backup copy of the data set and perform forward recovery. If you use CICSVR as your forward recovery utility, the backout is automatically retried for an RLS data set. For a non-RLS data set, use the SET DSNAME (...) 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 DSNAME (...) RETRY command to drive the backout retry.

NOTAPPLIC
The CVDA for CAUSE is not CONNECTION, RLSSERVER, or DATASET.

OPENERROR
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 in-doubt 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 in-doubt 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 in-doubt 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:

<table>
<thead>
<tr>
<th>Cause</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>CACHE</td>
<td>NOTAPPLIC</td>
</tr>
<tr>
<td>CONNECTION</td>
<td>INDOUBT</td>
</tr>
<tr>
<td>CONNECTION</td>
<td>RRINDOUBT</td>
</tr>
</tbody>
</table>

Chapter 2. System commands 415
INQUIRE UOWDSNFAIL

<table>
<thead>
<tr>
<th>Cause</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATASET</td>
<td>BACKUPNONBWO</td>
</tr>
<tr>
<td>DATASET</td>
<td>DELEXITERROR</td>
</tr>
<tr>
<td>DATASET</td>
<td>DATASETFULL</td>
</tr>
<tr>
<td>DATASET</td>
<td>DEADLOCK</td>
</tr>
<tr>
<td>DATASET</td>
<td>FAILEDBKOUT</td>
</tr>
<tr>
<td>DATASET</td>
<td>INDEXRECFULL</td>
</tr>
<tr>
<td>DATASET</td>
<td>LCKSTRUCFULL</td>
</tr>
<tr>
<td>DATASET</td>
<td>IOERROR</td>
</tr>
<tr>
<td>DATASET</td>
<td>OPENERROR</td>
</tr>
<tr>
<td>RLSSERVER</td>
<td>COMMITFAIL</td>
</tr>
<tr>
<td>RLSSERVER</td>
<td>RRCOMMITFAIL</td>
</tr>
<tr>
<td>RLSSERVER</td>
<td>RLSGONE</td>
</tr>
<tr>
<td>UNDEFINED</td>
<td>NOTAPPLIC</td>
</tr>
</tbody>
</table>

**RLSACCESS(cvda)**
returns a CVDA value indicating whether the data set was last opened in this CICS region in RLS or non-RLS mode. CVDA values are:

- **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.

Browse only function

The INQUIRE UOWENQ command can be used only in browse mode.

Description

The INQUIRE UOWENQ command retrieves information about enqueues. Enqueues are used by CICS 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

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.

Conditions: END, ILLOGIC, NOTAUTH, UOWNOTFOUND
INQUIRE UOWENQ

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
analyse 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
in-doubt failure, user ENQs are released while CICS enqueues are usually
converted to the retained state until the in-doubt 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 in a specific UOW, or
through all the enqueues currently in your system. See “Browsing resource
definitions” on page 21 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 prior to 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
INQUIRE UOWENQ

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 in-doubt 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 4 on page 421.
INQUIRE UOWENQ

**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 4 on page 421.

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.
INQUIRE UOWENQ

(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 in-doubt 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 4.

<table>
<thead>
<tr>
<th>TYPE</th>
<th>RESOURCE</th>
<th>QUALIFIER</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATASET</td>
<td>Data set name</td>
<td>Record identifier</td>
</tr>
<tr>
<td>EXECENQ</td>
<td>EXEC enqueue argument</td>
<td>None</td>
</tr>
<tr>
<td>EXECENQADDR</td>
<td>Address of EXEC enqueue argument</td>
<td>None</td>
</tr>
<tr>
<td>FILE</td>
<td>File name</td>
<td>Record identifier</td>
</tr>
<tr>
<td>TDQUEUE</td>
<td>TD queue name</td>
<td>FROMQ or TOQ</td>
</tr>
</tbody>
</table>
INQUIRE UOWENQ

Table 4. Data returned in RESOURCE and QUALIFIER (continued)

<table>
<thead>
<tr>
<th>TYPE</th>
<th>RESOURCE</th>
<th>QUALIFIER</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSQUEUE</td>
<td>TS queue name</td>
<td>None</td>
</tr>
</tbody>
</table>

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.

ILLOGIC
RESP2 values:
1 For INQUIRE UOWENQ START, means that a browse of this resource type is already in progress. For INQUIRE UOWENQ NEXT and INQUIRE UOWENQ END, means that an INQUIRE UOWENQ 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 UOWLINK

Retrieve information about a connection involved in a unit of work.

**Description**

The INQUIRE UOWLINK command retrieves information about a connection involved in a unit of work. The connection can be to a remote system, to a task-related user exit, or to a CFDT server.

If it is to a remote system, INQUIRE UOWLINK returns the netname of the connection, its sysid, and whether it is the coordinator or subordinate. If it is to a task-related user exit, INQUIRE UOWLINK returns the exit's entry name and qualifier. If it is to a CFDT server, INQUIRE UOWLINK returns the poolname.

**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 21 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 there is one). You can filter the UOW-links returned by specifying a value in the UOW field.

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.

**Conditions:** END, ILLOGIC, NOTAUTH, UOWLNOTFOUND, UOWNOTFOUND
INQUIRE UOWLINK

The browse guarantees that data for each UOW-link that exists prior to the first INQUIRE NEXT, and still exists after the last INQUIRE NEXT, is returned on exactly one INQUIRE NEXT call.

Options

HOST(data-area)
returns, for a TYPE of IIOP, the TCP/IP hostname, or a string containing the dotted decimal TCP/IP address, used to refer to the participant in the OTS transaction. This is useful for identifying the participant, especially when problems occur. This is a 255–character data-area. Strings of fewer than 255 characters are padded with blanks. Note that the UOW and the participant may belong to the same CorbaServer.

For TYPE values other than IIOP, HOST returns blanks.

LINK(data-area)
returns, for a TYPE value of CONNECTION, the 8-character netname of the remote system. For a TYPE value of RMI, LINK returns the entry name of the task-related user exit. For a TYPE value of CFTABLE, LINK returns the 8–character name of the coupling facility data table pool. For a TYPE value of IIOP, LINK returns blanks; the equivalent information for IIOP UOWLINKs is returned by the HOST attribute.

NETUOWID(data-area)
returns the 1-through 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:

- **APPC** Advanced Program to Program Communication.
- **IRC** InterRegion Communication. This is an MRO connection.
- **LU61** LUTYPE 6.1.
- **IIOP** OTS.
- **NOTAPPLIC** This is a connection of type CFTABLE or RMI.
- **RRMS** The UOW is coordinated by RRS/MVS.

RESYNCSTATUS(cvda)
returns a CVDA value indicating the resynchronization status of the connection. CVDA values are:

- **COLD** The connection has been cold started by the partner system. The partner can no longer coordinate any in-doubt 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 could be an RMI, an APPC single-session, an APPC synclevel 1 connection, or a CFDT server.
- **OK** The connection is operating normally. If there has been a failure of the partner system, the partner has been restarted and the connection is able to resynchronize the associated UOW.
INQUIRE UOWLINK

**STARTING**
The connection is being acquired, but the exchange lognames process has not yet completed.

**UNAVAILABLE**
The connection is not currently acquired.

**UNCONNECTED**
There is 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, 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 syncpoint coordinator for the UOW.

**SUBORDINATE**
This connection is to a syncpoint subordinate for the UOW.

**UNKNOWN**
The syncpoint 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 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:

**CFTABLE**
A connection to a CFDT server.

**CONNECTION**
A connection defined in a CONNECTION resource definition.

**OTS**
An OTS link.

**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.

**Conditions**

**END**
RESP2 values:
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.

UOWLNDFOUND
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.

Options

**URIMAP**(data-value

specifies the 8-character name of a URIMAP definition.

Description

The INQUIRE URIMAP command allows you to retrieve information about a particular URIMAP definition. The USAGE attribute of a URIMAP definition determines which other attributes are specified in that URIMAP definition, and sometimes determines the meaning of a particular attribute.

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.

Conditions: END, ILLOGIC, NOTAUTH, NOTFND

This command is threadsafe.
**ANALYZERSTAT** *(cvda)*
returns a CVDA value indicating whether the analyzer program associated with the TCPIPSERVICE definition is to be run. CVDA values are:

**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**.

**CERTIFICATE** *(data-area)*
returns a 32-character data area containing 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)**.

**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)**.

**ENABLESTATUS** *(cvda)*
returns a CVDA value indicating the status of this URIMAP definition. CVDA values are:

**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 re-enable 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 HFS file that forms a static response. This attribute is for **USAGE(SERVER)**.

**HOST** *(data-area)*
returns a 116-character data area containing the host component of the URI to which the URIMAP definition applies (for example, www.example.com). This attribute is for any usage type.

**HOSTCODEPAGE** *(data-area)*
returns a 10-character data area containing the 1-10 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).

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) or USAGE(PIPELINE).

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 which act on the service request from the client. This attribute is for USAGE(PIPELINE).

PROGRAM(data-area)
returns the 8-character name of the application program that composes an application-generated response to the HTTP request. This attribute is for USAGE(SERVER).

SCHEME(cvda)
returns a CVDA value indicating the scheme component of the URI. CVDA values are:

HTTP   HTTP without SSL.
HTTPS  HTTP with SSL.

This attribute is for any usage type.

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 is not specified, the URIMAP definition applies to a request on any inbound ports. This attribute is for USAGE(SERVER) or USAGE(PIPELINE).

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) or USAGE(PIPELINE).
REDIRECTTYPE(cvida)
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) or USAGE(PIPELINE). 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 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).

USAGE(cvida)
returns a CVDA value indicating the purpose of this URIMAP definition. CVDA values are:

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 invokes a Web service in CICS.

USERID(data-area)
returns the 8-character user ID under which the alias transaction is attached. This attribute is for USAGE(SERVER) or USAGE(PIPELINE).

WEBSERVICE(data-area)
returns the name of a Web service. This can be the 1-8 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 defines aspects of the run time 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.
INQUIRE UOWLINK

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 Transaction Server for OS/390, for retrieving information about journal volumes.

Description

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 Transaction Server 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 VTAM.

Options

**GRNAME**(data-area)

returns the 8-character generic resource group name under which this CICS region requests registration to VTAM, if it is using the generic resources facility of VTAM. 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, that a nonblank GRNAME value was specified). CVDA values are:

For more information about the use of CVDAs, see **CICS-value data areas (CVDAs)** on page 8.

Description

The INQUIRE VTAM command returns information about type and state of the connection between VTAM and your CICS system.
DEREGERROR
Deregistration was attempted but was unsuccessful, and there has been no attempt to reregister.

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 there has been no attempt to deregister.

REGISTERED
Registration was successful and there has been no attempt to deregister.

UNAVAILABLE
VTAM 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 VTAM. CVDA values are:

CLOSED
The connection between CICS and VTAM has not yet been established or has been terminated.

CLOSEFAILED
The connection is open but is not usable because a previous request to close the connection failed. You should retry the close request.

CLOSING
The connection between CICS and VTAM is in the process of closing.

FORCECLOSING
The connection between CICS and VTAM is in the process of closing following a SET VTAM FORCECLOSE command.

IMMCLOSING
The connection between CICS and VTAM is in the process of closing following a SET VTAM IMMCLOSE command.

OPEN
There is a connection between CICS and VTAM.

PSDINTERVAL(data-area)
returns the persistent session delay (PSD) interval, which is the length of time that sessions are held in recovery-pending state after a CICS failure. (See the PSDINT system initialization parameter in the CICS System Definition Guide for more information about this option.) There are two formats for the PSD interval:

- A composite (packed decimal format 0hmmss+, 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.
INQUIRE VTAM

(A value of zero means that sessions are not held after a failure, and may indicate that the VTAM in use is not at the level that supports persistent sessions.)

**PSDINTHRS**(\textit{data-area})
returns the hours component of the PSD interval, in fullword binary form (see the PSDINTERVAL option).

**PSDINTMINS**(\textit{data-area})
returns the minutes component of the PSD interval, in fullword binary form (see the PSDINTERVAL option).

**PSDINTSECS**(\textit{data-area})
returns the seconds component of the PSD interval, in fullword binary form (see the PSDINTERVAL option).

**Conditions**

**INVREQ**

RESP2 values:

1 VTAM 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.

Context

INQUIRE WEB returns information about the state of CICS Web support.

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 period of 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.
Use the INQUIRE WEBSERVICE command to retrieve information about an installed WEBSERVICE.

**Description**

Use the INQUIRE WEBSERVICE command to retrieve information about an installed WEBSERVICE.

**Browsing**

You can browse through all the WEBSERVICEs installed in your system by using the browse options (START, NEXT, and END) on INQUIRE WEBSERVICE commands. See [Browsing resource definitions](#) for general information about browsing, including syntax, exception conditions, and examples.

**Options**

**BINDING**(data-area)

Returns the WSDL binding represented by the WEBSERVICE. This binding is one of (potentially) many that appear in the WSDL file. The name can be up to 255 characters long.

**CONTAINER**(data-area)

Returns the name of the container used if PGMINTERFACE returns a value of CHANNEL. The name can be up to 16 characters long.

**ENDPOINT**(data-area)

Returns the endpoint URI of a remote WEBSERVICE. This is the endpoint URI specified in the WSDL file for a remote Web service. If a CICS application program is the service provider, then the ENDPOINT will be empty. The URI can be up to 255 characters long.

**LASTMODTIME**(data-area)

**Conditions**: NOTAUTH, NOTFND

---

**INQUIRE WEB**

Chapter 2. System commands 437
INQUIRE WEB

Returns the time, in milliseconds since 00:00 on January 1st 1900, that the deployed WSBind file on HFS was last updated. This 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 WEBSERVICEs (those installed by the CICS scanning mechanism), the value of LASTMODTIME is the timestamp of the HFS file pointed to by the WSBind definition, at the time the WEBSERVICE definition was last installed or updated.
- For statically-installed WEBSERVICEs (those installed from a CSD or by CREATE WEBSERVICE), the value of LASTMODTIME is the timestamp of the WSBind HFS file pointed to by the WEBSERVICE definition, at the time the WEBSERVICE was installed.

If you issue an INQUIRE WEBSERVICE command before a newly-installed or updated WEBSERVICE 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.

PGMINTERFACE(cvda)

Returns a CVDA indicating whether the CICS program that implements the Web service expects input in a channel or in a commarea:

CHANNEL
  The program expects input in a channel.

COMMAREA
  The program expects input in a commarea

PIPELINE(data-area)

Returns the name of the PIPELINE in which the WEBSERVICE 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 represents a remote Web service (that is, CICS is not the service provider), PROGRAM will be empty. The name can be up to 8 characters long.

STATE(cvda)

Returns a CVDA indicating the state of the WEBSERVICE:

DISCARDING
  A DISCARD command has been issued for the WEBSERVICE. The WEBSERVICE is quiescing before being discarded. It is not accepting new work, but is allowing currently-executing 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 WEBSERVICE is usable.
INQUIRE WEB

UNUSABLE
Copying of the WSBIND file on the shelf has failed, and the WEBSERVICE is unusable.

URIMAP(data-area)
Returns the name of a dynamically installed URIMAP if there is one that is associated with this WEBSERVICE. If the WEBSERVICE was not installed by performing the SCAN function on a PIPELINE resource, or if the WEBSERVICE represents a remote Web service, then the URIMAP will be 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 WEBSERVICE:

VALIDATION
Full validation is enabled.

DISABLED
Full validation is disabled.

WEBSERVICE(name)
Specifies the name of the WEBSERVICE 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.

Conditions

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 WEBSERVICE cannot be found.
INQUIRE WORKREQUEST

INQUIRE WORKREQUEST
Retrieve information about work requests in the local CICS region.

INQUIRE WORKREQUEST

\[\text{INQUIRE WORKREQUEST}(\text{data-value})\]

\begin{itemize}
\item \text{WORKTYPE(cvda)}
\item \text{CLIENTIPADDR(data-area)}
\item \text{CORBASENDER(data-area)}
\item \text{LISTENERPORT(data-area)}
\item \text{OAPPLID(data-area)}
\item \text{OTASK(data-area)}
\item \text{OTRANSID(data-area)}
\item \text{REQUESTID(data-area)}
\item \text{STACK(data-area)}
\item \text{TASK(data-area)}
\item \text{TRANSID(data-area)}
\item \text{TSYSTEM(data-area)}
\end{itemize}

Conditions: 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 WORKREQUEST command allows you 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 need not remain in the system after it has initiated a RequestProcessor; if there are no more requests to be received it terminates. A RequestProcessor may create RequestReceivers during the course of its processing - outbound and loopback requests.

INQUIRE WORKREQUEST enables you to inquire about a single work request in the local region, or to browse through all the work requests in the local region. The information returned by INQUIRE WORKREQUEST allows you to:

\begin{itemize}
\item determine which transactions are associated with this work request
\item correlate all transactions associated with this work request (for example, for accounting purposes)
\end{itemize}

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.
INQUIRE WORKREQUEST

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. See "Browsing resource definitions" on page 21 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)
   specifies the TCPIP address of the client that originated the request as a 15-character value.

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. There is one stack created for each request receiver, and the stack applies across the sysplex. The stack is decremented each time a bean that has incremented it terminates.

TASK(data-area)
   returns, in packed decimal form, the number of the local task (RequestProcessor).

TRANSID(data-area)
   specifies the local transaction id - 4 character

TSYSTEM(data-area)
   specifies the applid of the target CICS system as a 21-byte field. This field may contain one of the following:
   - the dotted decimal TCPIP address and port number of the target system
   - up to eight characters followed by blanks. In this case these 8-characters, or less, are the VTAM applid of the target system
INQUIRE WORKREQUEST

- If the field contains only a string of blank characters (spaces), the target is not CICS over MRO
- Another value. CICS does not know about any other possibilities. Any other value must be meaningful to other software at your installation which expects to work with the value obtained from this parameter of this CICS command

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 which 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 being performed. CVDA values are:

| I | IIOP | Specifies that the work is being executed for an IIOP request. |
| I | SOAP | Specifies that the work is being executed for a Web service request. |

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.
PERFORM CLASSCACHE

Initialize and terminate the shared class cache.

PERFORM CLASSCACHE

AUTOSTARTST(cvda)

sets the status of autostart for the shared class cache. The CVDA values are:

- ENABLED
  The shared class cache is started as soon as CICS receives a request to run a Java application in a JVM whose profile requires the use of the shared class cache.

Options

AUTOSTARTST(cvda)

sets the status of autostart for the shared class cache. The CVDA values are:

- ENABLED
  The shared class cache is started as soon as CICS receives a request to run a Java application in a JVM whose profile requires the use of the shared class cache.

Description

You can use the PERFORM CLASSCACHE command to initialize (start or reload) and terminate (phase out, purge or forcepurge) the shared class cache. 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) and change the JVM profile that is used for the master JVM (PROFILE option). The new settings override the size and JVM profile that were originally set by the system initialization parameters JVMCCSIZE and JVMCCPROFILE.

- 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 system initialization parameter JVMCCSTART.

When you change the size, JVM profile or autostart status of the shared class cache while CICS is running, subsequent CICS restarts use the most recent settings that you made using the PERFORM CLASSCACHE command (or the SET CLASSCACHE command), unless the system is INITIAL or COLD started, or the system initialization parameters are specified as overrides at startup. In these cases, the settings from the system initialization parameters are used.

Conditions: INVREQ, NOTAUTH
PERFORM CLASSCACHE

DISABLED

An explicit EXEC CICS PERFORM CLASSCACHE INITIALIZE(START) command (or the equivalent CEMT command) is required to start the shared class cache. 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 whose profile requires the use of the shared class cache, the request fails.

CACHESIZE(data-value)

is a doubleword binary value. If you are using the INITIALIZE(START) option, CACHESIZE specifies the size, in bytes, of the shared class cache that is to be started. If you are using the INITIALIZE(RELOAD) option, CACHESIZE specifies the size, in bytes, of the new shared class cache that is to be loaded.

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:

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, CACHESIZE and PROFILE are used in the creation of the 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, CACHESIZE and PROFILE are used in the creation of the cache. If newer versions of Java classes are available to be loaded, the new class cache uses them. However, worker 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 and the old versions of the Java classes until the new shared class cache is ready. When the new shared class cache is ready, subsequent requests for worker JVMs are given a worker JVM that uses the new cache. These new worker JVMs are started as they are requested by applications, and they replace the worker JVMs that are using the old shared class cache. The worker 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 worker JVMs that are dependent on it have been terminated.

PROFILE(data-value)

If you are using the INITIALIZE(START) option, PROFILE specifies the eight-character name of the JVM profile that will be used for the master JVM that starts the shared class cache. If you are using the INITIALIZE(RELOAD) option, PROFILE specifies the eight-character name of the JVM profile that will be used for the master JVM that loads the new shared class cache.

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 HFS file name.

If the START or RELOAD command is successfully executed but the master JVM fails to start, CICS retains the JVM profile name that you specify, and
PERFORM CLASSCACHE

displays it as the JVM profile for the master JVM. Next time you issue the command to start or reload the shared class cache, the new JVM profile will be used, unless you change it again using the PROFILE option.

TERMINATE(cvda)
specifies that the shared class cache is to be stopped and deleted. All worker JVMs using the shared class cache are terminated. No more JVMs can use the shared class cache, and it is deleted when all the worker JVMs that were dependent on 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 EXEC CICS 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:

PHASEOUT
All worker JVMs using the shared class cache are marked for deletion. The JVMs are actually deleted when they finish running their current Java programs. The shared class cache is deleted when all the worker JVMs that were dependent on it have been terminated.

PURGE
All tasks using worker JVMs dependent on the shared class cache are terminated by the SET TASK PURGE mechanism, and the worker JVMs are terminated. The shared class cache is deleted when all the worker JVMs that were dependent on it have been terminated.

FORCEPURGE
All tasks using worker JVMs dependent on the shared class cache are terminated by the SET TASK FORCEPURGE mechanism, and the worker JVMs are terminated. The shared class cache is deleted when all the worker JVMs that were dependent on 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 which is already stopped.
6 START was specified for a shared class cache which is not stopped.
7 RELOAD was specified for a shared class cache which is not started.
8 JVM profile name contained invalid characters, or started with a blank followed by valid characters.

Chapter 2. System commands 445
PERFORM CLASSECACHE

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.

<table>
<thead>
<tr>
<th>PERFORM CORBASERVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORBASERVER(data-value)</td>
</tr>
<tr>
<td>ACTION(cvda)</td>
</tr>
<tr>
<td>PUBLISH</td>
</tr>
<tr>
<td>RETRACT</td>
</tr>
<tr>
<td>SCAN</td>
</tr>
</tbody>
</table>

**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 the hierarchical file store (HFS) 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 HFS directories, such as those specified on the JVMPROPS, LIBPATH, and CLASSPATH parameters.

**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:
**PERFORM CORBASERVER**

**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.

**Notes:**

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 DELETSHIPPED

Delete inactive shipped terminal definitions.

PERFORM DELETSHIPPED

Description

The PERFORM DELETSHIPPED 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 DELETSHIPPED command, and you can set these values with the SET DELETSHIPPED 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.

**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:
PERFORM DJAR

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

PERFORM DUMP DUMPCODE(data-value)

TITLE(data-area)—TITLELENGTH(data-value)

CALLER(data-value)—CALLERLENGTH(data-value)

DUMPID(data-area)

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 using default values. See the INQUIRE SYSDUMPCODE command for more information about this process and 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. The CICS Messages and Codes manual 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.
PERFORM DUMP

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 will build temporary entries, as explained above.

**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 to 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 CALLEERLENGTH 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.
PERFORM DUMP

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.

PERFORM ENDAFFINITY

\[ \text{NETNAME}(\text{data-value}) \]
\[ \text{NETID}(\text{data-value}) \]

Conditions: INVREQ

Description

Where CICS is a member of a VTAM generic resource group, the PERFORM ENDAFFINITY command instructs VTAM to end an affinity owned by CICS, whether or not the connection has been deleted. If the connection has not been deleted, it must be out of service and have no recovery information outstanding (that is, its RECOVSTATUS must be NORECOVDATA).

Generic resources and affinities are described in the CICS Intercommunication Guide.

Note: There is no facility in VTAM 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 you must end explicitly, CICS issues message DFHZC0177. This message gives the NETNAME and NETID to be passed to VTAM.

If a request to end an affinity is rejected by VTAM because no such affinity exists, CICS issues message DFHZC0181. This may mean either that your program specified an incorrect NETNAME or NETID, or that it (or CICS) was wrong in supposing that an affinity existed.

Options

\text{NETID}(\text{data-value})

specifies the name by which the network containing the remote LU is known to VTAM.

If you do not specify a NETID, CICS takes the value from the installed connection, if it exists. If you do not specify a NETID and the connection does not exist, the command fails.

\text{NETNAME}(\text{data-value})

specifies the APPLID of the remote LU. If the connected LU is a member of a generic resource, you must specify its member name, not the generic resource name.

Conditions

INVREQ

RESP2 values:

25 The connection is still in service.

26 There may be recovery information outstanding for the connection. RECOVSTATUS has a value other than NORECOVDATA.
PERFORM ENDAFFINITY

32 See message DFHZC0178. VTAM could not end the affinity for a reason other than 35 (NOTFOUND) or 36 (SESSIONS ACTIVE).
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 VTAM could not find an affinity for the values input.
36 VTAM could not end the affinity because the connection had some sessions active.
37 See message DFHZC0176. A VTAM error prevented the CHANGE ENDAFFIN macro being carried out.
PERFORM PIPELINE

PERFORM PIPELINE

Initiate a scan of the Web service binding files that are associated with a PIPELINE.

PERFORM PIPELINE

\[ \text{PERFORM PIPELINE}\left(\text{data-value}\right) \]
\[ \quad \text{ACTION (cvda)} \]
\[ \quad \text{SCAN} \]

Conditions: DUPRES, INVREQ, NOTAUTH, NOTFND

Description

Use the PERFORM PIPELINE command to initiate a scan of the Web service binding directory that is specified in the WSBIND attribute of the PIPELINE definition. If the WSBIND attribute is not specified, there is nothing to scan, and control returns to your program.

If the directory location specified is valid, CICS examines the Web service binding files in the directory to determine if they should be installed into the system:

- CICS installs any files it finds that have not been installed already.
- If a file has been installed already, but the file in the directory is newer than the one currently in use, the one that is in use is discarded, and the newer file is installed in its place.

If, for any reason, CICS fails to install an individual Web service binding file, processing continues with the remaining files in the directory. When the scan completes, the PIPELINE is available for use with whichever of the binding files were installed successfully.

Options

\[ \text{PIPELINE}\left(\text{data-value}\right) \]

specifies the 8–character name of the PIPELINE.

\[ \text{ACTION (cvda)} \]

\[ \text{SCAN} \]

Specifies a CVDA value indicating the action to be taken on the PIPELINE.

CVDA values are:

\[ \text{SCAN} \]

Scan the PIPELINE’s Web service binding directory

Conditions

\[ \text{DUPRES} \]

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.
PERFORM 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

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.

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.
PERFORM SECURITY REBUILD

**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.

**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 VTAM 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.

**Conditions**: INVREQ, NOTAUTH, TRANSIDERR
PERFORM SHUTDOWN

This option applies to immediate shutdowns only. If you specify NORESTART without IMMEDIATE, CICS performs an immediate shutdown as if you specified the command as PERFORM SHUTDOWN IMMEDIATE NORESTART.

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 an 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:
PERFORM SHUTDOWN

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.
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

Conditions: IOERR, NOTAUTH, NOTFND

Description
system options that control the recording of these statistics at intervals. (See the
discussion of interval statistics in the SET STATISTICS command on page "SET
STATISTICS options - ENDOFDAYSECS" on page 567.)

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 wish, 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.

In addition, you can reset the counts when you use this option (see the
RESETNOW option).

AUTOINSTALL
records global statistics on the automatic installation of terminal definitions.

BEAN
records specific statistics for all enterprise beans installed in the CICS region.

CONNECTION
records specific statistics for all 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.

ENQUEUE
records global statistics for the enqueue manager.

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.

JOURNALNAME
records specific statistics for all journals installed in the CICS region. This
PERFORM STATISTICS RECORD

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.

**JVMPOOL**
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.

**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.

**PIPELINE**
records statistics related to a PIPELINE.

**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).

**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.
PERFORM STATISTICS RECORD

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
records global VTAM statistics for the CICS region.

WEBSERVICE
records statistics related to a WEBSERVICE 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:
PERFORM STATISTICS RECORD

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" on page 46 for additional information.
```

The RESP2 value $n$ identifies the last type to fail in this way, as follows:

<table>
<thead>
<tr>
<th>$n$</th>
<th>Resource type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AUTOINSTALL</td>
</tr>
<tr>
<td>2</td>
<td>CONNECTION</td>
</tr>
<tr>
<td>3</td>
<td>DISPATCHER</td>
</tr>
<tr>
<td>6</td>
<td>FILE</td>
</tr>
<tr>
<td>8</td>
<td>JOURNALNUM and JOURNALNAME</td>
</tr>
<tr>
<td>10</td>
<td>LSRPOOL</td>
</tr>
<tr>
<td>11</td>
<td>MONITOR</td>
</tr>
<tr>
<td>12</td>
<td>PROGRAM</td>
</tr>
<tr>
<td>13</td>
<td>STATS</td>
</tr>
<tr>
<td>14</td>
<td>STORAGE</td>
</tr>
<tr>
<td>15</td>
<td>SYSDUMP</td>
</tr>
<tr>
<td>16</td>
<td>TABLEMGR</td>
</tr>
<tr>
<td>18</td>
<td>TCLASS, TRANCLASS</td>
</tr>
<tr>
<td>19</td>
<td>TDQUEUE</td>
</tr>
<tr>
<td>20</td>
<td>TERMINAL</td>
</tr>
<tr>
<td>21</td>
<td>TRANDUMP</td>
</tr>
<tr>
<td>22</td>
<td>TRANSACTION</td>
</tr>
<tr>
<td>23</td>
<td>TSQUEUE</td>
</tr>
<tr>
<td>24</td>
<td>VTAM</td>
</tr>
<tr>
<td>25</td>
<td>FEPI</td>
</tr>
<tr>
<td>26</td>
<td>PROGAUTO</td>
</tr>
<tr>
<td>28</td>
<td>ENQUEUE</td>
</tr>
<tr>
<td>29</td>
<td>RECOVERY</td>
</tr>
<tr>
<td>30</td>
<td>STREAMNAME</td>
</tr>
<tr>
<td>31</td>
<td>DB2</td>
</tr>
<tr>
<td>32</td>
<td>TCPIPSERVICE</td>
</tr>
<tr>
<td>33</td>
<td>TCPIP</td>
</tr>
<tr>
<td>34</td>
<td>REQUESTMODEL</td>
</tr>
<tr>
<td>35</td>
<td>CORBASERVER</td>
</tr>
<tr>
<td>36</td>
<td>JVMPOOL</td>
</tr>
<tr>
<td>37</td>
<td>JVMPROFILE</td>
</tr>
<tr>
<td>38</td>
<td>BEAN</td>
</tr>
<tr>
<td>39</td>
<td>JVMPROGRAM</td>
</tr>
<tr>
<td>40</td>
<td>MQCONN</td>
</tr>
<tr>
<td>41</td>
<td>URIMAP</td>
</tr>
<tr>
<td>42</td>
<td>WEBSERVICE</td>
</tr>
<tr>
<td>43</td>
<td>PIPELINE</td>
</tr>
</tbody>
</table>
Determine the disposition of “in doubt” units of work.

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 in-doubt 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 in-doubt units of work at once, it should specify PARTIAL on the RESYNC command. If it does not, CICS discards disposition information for all the in-doubt units of work that are not in the supplied list, but are part of the resource manager’s resynchronization set. 3

3. 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 in-doubt 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 in-doubt unit of work) of the address-list.

**PARTIAL**

specifies that CICS is to retain in-doubt resolution data for the UOWs (for this resource manager) that are not passed in the in-doubt 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.
RESYNC ENTRYNAME

If PARTIAL is not specified, CICS discards resolution data for any UOWs not passed in the in-doubt list, but which are part of this resource manager’s resynchronization set. 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 AUTOINSTALL

Set autoinstall values.

**Description**

The SET AUTOINSTALL command lets you change some of the values that control the automatic installation (autoinstall) of VTAM terminals, APPC sessions, virtual terminals (bridge facilities) used by the 3270 bridge mechanism, and MVS consoles in a CICS region.

**Options**

**AIBRIDGE**

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**

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 ¬ (logical not) symbol as the first character.

**NOAUTO**

Autoinstall for consoles is not allowed.

**MAXREQS**

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.

**Conditions:** INVREQ, NOTAUTH, PGMIDERR
SET AUTOINSTALL

**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 VTAM 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
  #  SET BRFACILITY—(data-area)—
  TERMSTATUS(cvda)—
  RELEASED

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-area)
  - 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.
SET BR FACILITY

NOTFOUND
RESP2 values:
1 The specified bridge facility could not be found.
SET CLASSCACHE

Set the status of autostart for the shared class cache.

### Options

**AUTOSTARTST**(cvda)

specifies the status of autostart for the shared class cache. The CVDA values are:

- **ENABLED**
  
  If the shared class cache has been stopped or has not yet been started on this CICS execution, the shared class cache is started as soon as CICS receives a request to run a Java application in a JVM whose profile requires the use of the shared class cache.

- **DISABLED**
  
  If the shared class cache has been stopped or has not yet been started on this CICS execution, an explicit EXEC CICS PERFORM CLASSCACHE INITIALIZE(START) command (or the equivalent CEMT 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
SET CLASSCACHE

run a Java application in a JVM whose profile requires the use of the shared class cache, the request fails.

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

Description

The SET CONNECTION command allows you to change some of the attributes that define a 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.

There are two main types of system connection:

- **Inter region communication (IRC) connections**, which use the CICS interregion communication program (DFHIRP) to establish a connection between two MRO partners. An IRC connection can exist between two CICS regions on the same MVS image or within the same MVS/ESA sysplex.

  A special form of IRC connection, used by the external CICS interface (EXCI) can exist between a CICS region and a non-CICS client program running in MVS, for example an MVS batch or TSO program. An EXCI connection connects the client program to a CICS region running in the same MVS image or within the same MVS/ESA sysplex.

- **Intersystem communication (ISC) connections**, between CICS and any other system which supports VTAM APPC or LUTYPE6.1 communications. For example, ISC connections can exist between CICS regions running in different MVS/ESA sysplexes or on different operating system platforms, between CICS and any APPC device, and between CICS and IMS.

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.

Conditions: INVREQ, IOERR, NORMAL, NOTAUTH, SYSIDERR
The process of acquiring and releasing APPC sessions 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 data identifier (dataid) prefix DF. If temporary storage dataids 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)**

is retained only for compatibility purposes. You should use CONNSTATUS in new applications.

**AFFINITY(cvda) (APPC and LU6.1 only)**

specifies, where CICS is a member of a VTAM generic resource group, that VTAM 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 VTAM 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 VTAM 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](#).
SET CONNECTION

**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) (VTAM 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 in-doubt 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 in-doubt 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.

**Notes:**

1. **SET CONNECTION NOTPENDING, SET CONNECTION NORECOVDATA, and SET CONNECTION UOWACTION** are mutually exclusive. For advice on which command to use, see the description of the UOWACTION option on page "SET CONNECTION options - SERVSTATUS" on page 484.

2. **NOTPENDING** has no effect on MRO connections to pre-CICS Transaction Server for z/OS systems. However, the cold start of a pre-CICS Transaction Server for z/OS MRO partner causes the SET CONNECTION NOTPENDING function to be performed automatically, session by session.

**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>
<thead>
<tr>
<th>Remote delete AIDs</th>
<th>CRSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote scheduler AIDs</td>
<td></td>
</tr>
<tr>
<td>LU6.2 service manager 1 AIDs</td>
<td>CLS1</td>
</tr>
<tr>
<td>LU6.2 service manager 3 AIDs</td>
<td>CLS3</td>
</tr>
<tr>
<td>Remote schedule PURGE AIDs</td>
<td>CRSQ</td>
</tr>
<tr>
<td>Resource manager resync AIDs</td>
<td>CRSY</td>
</tr>
<tr>
<td>Autostart terminal delete AIDs</td>
<td>CATD</td>
</tr>
<tr>
<td>Restart terminal delete AIDs</td>
<td>CATR</td>
</tr>
</tbody>
</table>

Message DFHTF0101 is written to the 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 5 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 rollback processing), CICS might terminate abnormally.

For in-doubt and shunted UOWs, FORCEPURGE has no effect.

**Note:** To force shunted UOWs, the operator must issue SET CONNECTION COMMIT, BACKOUT, or FORCE following a
SET CONNECTION

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 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
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 in-doubt 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 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 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).
SET CONNECTION

- 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 in-doubt 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:
SET CONNECTION

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.

FORCE
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 in-doubt 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 in-doubt 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 in-doubt 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) (VTAM only)
specifies whether the VTAM control component of CICS is to trace activity on the sessions associated with this connection. CVDA values are:

NOZCPTRACE
VTAM ZCP tracing is not to be carried out.

ZCPTRACE
VTAM ZCP tracing is to be carried out.

Conditions

INVREQ
RESP2 values:
1 ACQSTATUS|CONNSTATUS specified for a non-APPC connection.
SET 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 VTAM.

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 ZCPTTRACING has an invalid CVDA value.

14 EXITTRACING|ZCPTTRACING specified for a non-VTAM connection or VTAM 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. VTAM could not end the affinity for a reason other than 35 (NOTFOUND) or 36 (SESSIONS ACTIVE).

35 VTAM could not find an affinity for this connection.

36 VTAM could not end the affinity because the connection had some sessions active.

37 See message DFHZC0176. A VTAM 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:
SET CONNECTION

10  Unexpected error.

NORMAL  
RESP2 values:
58  AIDs are successfully canceled.
59  No AIDs are 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 connection could not be found.
SET CORBASERVER

Set various attributes of a specified CorbaServer.

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.

Description

The SET CORBASERVER command allows you to:
- Enable or disable a CorbaServer
- Turn the CorbaServer’s automatic publishing feature on or off
- Set the session bean timeout value

Options

**AUTOPUBLISH**(cvda)
specifies whether the contents of a deployed JAR file should be automatically published to the namespace when the DJAR definition is successfully installed into this CorbaServer.

**NOAUTO**
The contents of the deployed JAR file will not be automatically published to the namespace.

**AUTOPUB**
The contents of the deployed JAR file will be automatically published to the namespace.

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.
SET CORBASERVER

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, prior to 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:
SET CORBASERVER

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.

**Conditions:** NORMAL, NOTAUTH, NOTFND, INVREQ

This command is threadsafe.

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.

**Description**

The `SET DB2CONN` command also specifies the attributes of the pool and command threads.

**Note:** This command cannot be used in a remote program that is linked by a distributed program link command.
Options

ACCOUNTREC

Specifies the minimum amount of DB2 accounting required for transactions using pool threads. The specified minimum may 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 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 what id should be used for security checking for pool threads. If AUTHID is specified, AUTHTYPE may not be specified.

AUTHTYPE
specifies the type of ID that can be used for pool threads. 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.

<table>
<thead>
<tr>
<th>IDs passed to DB2</th>
<th>How DB2 interprets values</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS sign-on user ID</td>
<td>Represents the primary DB2 authorization ID.</td>
</tr>
<tr>
<td>(USERID)</td>
<td></td>
</tr>
<tr>
<td>RACF-connected</td>
<td>If the RACF list of group options is not active, DB2 uses the group name supplied by the</td>
</tr>
<tr>
<td>group name</td>
<td>CICS attachment facility as the secondary DB2 authorization ID. If the RACF list of group</td>
</tr>
<tr>
<td></td>
<td>options is active, DB2 ignores the connected group name supplied by the CICS attachment</td>
</tr>
<tr>
<td></td>
<td>facility, but the value appears in the DB2 list of secondary DB2 authorization IDs.</td>
</tr>
</tbody>
</table>

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.
SET DB2CONN

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 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  this parameter is valid only with CONNECTST when setting the CICS DB2 connection NOTCONNECTED. CVDA values are:

FORCE  similar to issuing DSNC 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  specifies which id should be used for security checking when using command threads. If COMAUTHID is specified, COMAUTHType may not be specified.

COMAUTHTYPE  specifies the type of id that can be used for security checking when using command threads. If COMAUTHType is specified, COMAUTHID may not be specified. CVDA values are:

CGROUP  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.
CICS sign-on user ID (USERID)  
Represents the primary DB2 authorization ID.

<table>
<thead>
<tr>
<th>IDs passed to DB2</th>
<th>How DB2 interprets values</th>
</tr>
</thead>
<tbody>
<tr>
<td>RACF-connected group name</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 COMAUTHTYPE(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 userid 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 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 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 COMAUTHTYPE(CUSERID) and COMAUTHTYPE(CGROUP).

COMTHREADLIM  
specifies the current maximum number of command threads that the CICS DB2 attachment allows active before requests overflow to the pool.

CONNECTERROR  
returns how the fact that CICS is not connected to DB2 because the adaptor is in 'standby mode' is reported back to an application that has issued a SQL request. CVDA values are:

ABEND  
The application is abended with abend code AEY9.

SQLCODE  
The application receives a -923 SQLCODE.
SET DB2CONN

CONNECTST
sets the status of the CICS DB2 connection; that is, to start or stop the CICS DB2 connection. CVDA values are:

CONNECTED
This is equivalent to issuing DSNC 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 dependant 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 will connect 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
specifies the name (up to four characters) of a data sharing group of DB2 subsystems. CICS attempts to connect to any active member of this group, using group attach. 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 needs to 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
specifies the 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 needs to 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
specifies the first transient data destination to which unsolicited messages from the CICS DB2 attachment are sent.

MSGQUEUE2
specifies the second transient data destination to which unsolicited messages from the CICS DB2 attachment are sent.
MSGQUEUE3
specifies the third transient data destination to which unsolicited messages from
the CICS DB2 attachment are sent.

NONTERMREL
specifies whether or not non-terminal transactions release threads for reuse at
intermediate syncpoints. CVDA values are:

RELEASE
non-terminal transactions release threads for reuse at intermediate
syncpoints.

NORELEASE
non-terminal transactions do not release threads for reuse at
intermediate syncpoints.

PLAN
specifies the name of the plan to be used for all threads in the pool. If PLAN is
specified, PLANEXITNAME may not be specified.

PLANEXITNAME
specifies the 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 may not
be specified.

PRIORITY
specifies the priority of the pool thread TCBs relative to the CICS main TCB
(QR TCB). If CICS is connected to DB2 Version 6 or later, the thread TCBs are
CICS open L8 TCBs. If CICS is connected to DB2 Version 5 or earlier, the
thread TCBs are private TCBs created by the CICS-DB2 Attachment Facility.
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
specifies in minutes the length of time of the protected thread purge cycle. The
default is 0,30; that 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. Hence if the purge cycle is set to 30 seconds after it is released,
a protected thread is purged 30 - 60 seconds after it is released. An
unprotected thread is terminated when it is released (at syncpoint or end of
task) if there are no other transactions waiting for a thread on that DB2ENTRY.

PURGECYCLES
specifies in seconds the length of time of the protected thread purge cycle. The
default is 0,30; that 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. Hence if the purge cycle is set to 30 seconds after it is released,
a protected thread is purged 30 - 60 seconds after it is released. An
unprotected thread is terminated when it is released (at syncpoint or end of
task) if there are no other transactions waiting for a thread on that DB2ENTRY.
RESYNCMEMBER
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 will occur 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.

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
specifies the 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
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 subsequently fails the CICS DB2 attachment reverts again to standby mode and subsequently reconnects to DB2 when it comes up again.

STATSQUEUE
specifies the transient data destination for CICS DB2 attachment statistics produced when the CICS DB2 attachment is shutdown.

TCBLIMIT
specifies 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
SET DB2CONN

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
specifies 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
specifies whether or not 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 userid under which SET is being executed is authorized to act on behalf of the userid 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/exit is determined using the new rules.

Conditions

NORMAL
RESP2 values:

38  Waiting for DB2 (this may 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.
SET DB2CONN

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 NTERMREL 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
SET DB2CONN

54  Invalid RESYNCHMEMBER option
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.

Conditions: NOTAUTH, NOTFND, INVREQ

This command is threadsafe.

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
SET DB2ENTRY

Description

The entry is identified by the name it was defined with in CEDA. For RCTs migrated to the CSD, the name is the name of the first transaction on the DSNCRCT TYPE=ENTRY statement.

Options

ACCOUNTREC

Specifies the minimum amount of DB2 accounting required for transactions using pool threads. The specified minimum may 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 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.

<table>
<thead>
<tr>
<th>IDs passed to DB2</th>
<th>How DB2 interprets values</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS sign-on user ID (USERID)</td>
<td>Represents the primary DB2 authorization ID.</td>
</tr>
</tbody>
</table>
SET DB2ENTRY

<table>
<thead>
<tr>
<th>IDs passed to DB2</th>
<th>How DB2 interprets values</th>
</tr>
</thead>
<tbody>
<tr>
<td>RACF-connected</td>
<td>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.</td>
</tr>
</tbody>
</table>

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.
SET DB2ENTRY

**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 MSGQUEUEEn 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). If CICS is connected to DB2 Version 6 or later, the thread TCBs are CICS open L8 TCBs. If CICS is connected to DB2 Version 5 or earlier, the thread TCBs are private TCBs created by the CICS-DB2 Attachment Facility. 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.
### SET 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 whilst 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
SET DB2ENTRY

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.

**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
SET DB2TRAN

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 DELETISHED

Change the system settings that control automatic deletion of shipped terminal definitions.

**Description**

The SET DELETISHED 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 the CICS Intercommunication Guide and the CICS 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 `0hhmmss+`, 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)`

**Conditions:** INVREQ, NOTAUTH
SET DELETSHIPPED

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** command,
described on page "PERFORM DELETSHIPPED" on page 450.

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 INTERVALMINS). 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 INTERVALMINS value is invalid.
4 The INTERVALSECS value is invalid.
5 The IDLE value is invalid.
SET DELETISHED

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.

### Description

The SET DISPATCHER command allows you 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. Table 3 on page 318 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 there are tasks 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 there are tasks 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 to 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 to 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 executing 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 500–2700000. 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 DISPATCHER SCANDELAY command.

Conditions

INVREQ
RESP2 values:
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.
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 Version 6 or later).
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 DSNAME

Change information relating to an external data set, including actions that apply to all UOWs that access this dataset.

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 in-doubt failures).
- Force any UOWs that are shunted due to in-doubt failures, and which have updated this data set, to complete.
- Purge shunted UOW log records that hold retained locks (other than those due to in-doubt 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 more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

For information about shunted UOW log records, see the CICS Recovery and Restart Guide.
The options and CVDA values 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, you should 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. Note that `QUIESCESTATE` 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 [z/OS V1R4.0 Security Server RACF Security Administrator's Guide](https://www.ibm.com/docs/en/zos/2.4.0).

**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 523 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:**

1. This option does not apply to shunted *in-doubt* UOWs. You should try to resolve the shunted in-doubt 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).

2. 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 in-doubt 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(cva)** (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(cva)** (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 DSNAME 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 the QUIESTIM system initialization parameter in the CICS 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 DSNAME operand of the CEDA DEFINE FILE command.

**QUIESCESTATE(cva)** (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 DSNAME UNQUIESCED command restores the file state to ENABLED, provided it was set UNENABLED by a QUIESCED or IMMQUIESCED action, but
SET DSNAME

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 either:
• Successfully committed, or
• Successfully backed out, or
• Shunted because of an in-doubt 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 QUIESCED 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 UNENABLED 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 in-doubt UOWs. CVDA values are:
SET DSNAME

BACKOUT
All shunted in-doubt UOWs that hold locks on this data set should be backed out.

COMMIT
All shunted in-doubt UOWs that hold locks on this data set should be committed.

FORCE
All shunted in-doubt 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 DSNAME 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.
SET DSNAME

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 cancelled 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 cancelled 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

```assembly
* 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('00'X)
*   Remove DSN block from storage
  EXEC CICS SET DSNAME(TEMPDSN) REMOVE
*
*   Return and end
*
RETURN DS 0H
EXEC CICS RETURN
END
```

PL/I example

```pli
*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
```

Chapter 2. System commands
/*
/* 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.

**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**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INITIALDDS(data-value)</td>
<td>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 <strong>CICS System Definition Guide</strong> for a description of this option.)</td>
</tr>
<tr>
<td>OPENSTATUS(cvda)</td>
<td>Specifies actions to be taken on the transaction dump data sets. CVDA values are:</td>
</tr>
<tr>
<td></td>
<td>CLOSED  The active CICS dump data set is to be closed.</td>
</tr>
<tr>
<td></td>
<td>OPEN  The active CICS dump data set is to be opened.</td>
</tr>
</tbody>
</table>

For more information about the use of CVDAs, see [CICS-value data areas (CVDAs)](page 8).
SET DUMPDS

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**

<table>
<thead>
<tr>
<th>ENQMODEL(data-value)</th>
<th>STATUS(cvda)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DISABLED</td>
</tr>
<tr>
<td></td>
<td>ENABLED</td>
</tr>
</tbody>
</table>

**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:
SET ENQMODEL

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 FILE

Change attributes of a VSAM or BDAM file, including files that refer to CICS shared data tables and coupling facility data tables.

Conditions: FILENOTFOUND, INVREQ, IOERR, NOTAUTH
For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.

Note: This command replaces the SET DATASET command. The DATASET keyword is supported by the translator as a synonym for FILE, but you
should 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. All changes, other than to close and disable the file, require that the file be in a CLOSED state, with an ENABLESTATUS of either DISABLED or UNENABLED, and they 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. You can use this capability to 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 within a single MVS image to accessing a coupling facility data table within a Parallel Sysplex.

If a coupling facility data table already 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 existence 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
EXEC CICS SET FILE(data-value)—DISABLED—NOWAIT
EXEC CICS SET FILE(data-value)—DISABLED—FORCE

Conditions: FILENOTFOUND, INVREQ, NOTAUTH
```
The SET FILE command allows you 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 are allowed to change are the ENABLESTATUS and the OPENSTATUS. Attempting to specify any other attribute when there are retained locks causes an INVREQ condition to be raised.

Options

ADD(cvda)
specifies whether new records are to be added to the file. CVDA values are:

ADDABLE
New records are to be added to the file.

NOTADDABLE
New records are not to be added to the file.

BROWSE(cvda)
specifies whether the file is to be browsable. CVDA values are:

BROWSABLE
The file is to be browsable.

NOTBROWSABLE
The file is not to be browsable.

BUSY(cvda)
specifies what CICS is to do if the file is in use when you issue the SET command. The BUSY option is valid only for requests to SET the file DISABLED or CLOSED, and is ignored for any other request. CVDA values are:

FORCE
All tasks using the file are abended, the file is immediately DISABLED or CLOSED, and control returns to the issuing application.
SET FILE

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 is to wait until all activity on the file has quiesced before setting the file DISABLED or CLOSED. CICS then returns control to the application that is issuing this command. WAIT is the default.

Note: Closing a file using the FORCE option causes tasks of any current users of the file to be terminated 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 terminate abnormally. For this reason, closing files using the FORCE option should be restricted to exceptional circumstances.

CFDTPOOL(data-area) (CFDT only)
specifies the name of the pool in which the coupling facility data table resides. You can specify the CFDT pool name 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.

DELETE(cvda) (VSAM only)
specifies whether records can be deleted from the file. CVDA values are:

DELETABLE
Records are deletable from the file.

NOTDELETABLE
Records are not deletable 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:

OLD
The disposition value is to be OLD.

SHARE
The disposition value is to be 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 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 specified 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).

Note: When you add a data set name to a file definition for a coupling facility data table, LSR pool size calculations may be involved when the file is opened. This occurs when the file refers to an LSRPOOL that CICS builds using default values, and the first data set using the LSR pool is opened to load the table. CICS issues message DFHFC0208 indicating 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 will not have been
included in the LSR pool calculation. This means the existing LSR pool
may 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
reference 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 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. This does not apply to files operating
in RLS mode.

EMPTYSTATUS(cxda) (VSAM only)
specifies whether the data set is to be emptied when a file that references it is
next opened. This 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:
1. If you specify EMPTYREQ for a nonreuseable data set, CICS
accepts it, but a subsequent attempt to open the file fails.
2. 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.
3. 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.
4. 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.
5. 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(cxda)
specifies whether application programs can access the file. CVDA values are:

DISABLED
The file is unavailable for access by application programs.

ENABLED
The file is available for access by application programs.
**SET FILE**

**EXCLUSIVE(cvda) (BDAM only)**
specifies whether records on this file should be placed under exclusive control when a read for update is issued. CVDA values are:

**EXCTL**  
Records on this file are to be under exclusive control.

**NOEXCTL**  
Records on this file are not to be under exclusive control.

**FILE(data-value)**
specifies the 8-character file name defined in the file control table (FCT).

**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 keylength, specify a value in the range 1 through 16. To clear a keylength (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:

**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 at a later date.

**LSRPOOLID(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 1–8.

If the file is not to 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 table 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 the data table for this file can hold. Using this parameter enables you 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 to 99 999 999. 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 between 5 and 10% more than the maximum number of records
SET FILE

that the table is expected to contain. This 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 apparently has fewer records than the MAXNUMRECS limit specifies.

OPENSTATUS(cvda)
specifies whether the file is to be open or closed. CVDA values are:

CLOSED
The file is to be closed.
The close request is deferred until all UOWs that hold repeatable read locks reach their syncpoint.

Note: 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 from being accessed through another file or by other CICS regions.

You can use the MVS MODIFY command to issue CFDT server commands that:
- Set the table unavailable (MODIFY server-name,SET TABLE=tablename,AVAILABLE=NO) so that no other files can issue opens against it
- 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 to be 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 to be closed, the task issuing the close must commit any prior changes to that data set, or the request is rejected by file control.

READ(cvda)
specifies whether records can be read from the file. CVDA values are:

NOTREADABLE
Records are not to be readable from the file.

READABLE
Records are to be readable from the file.

READINTEG(cvda)
specifies the default level of read integrity for the file. CVDA values are:
SET FILE

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.

Note: 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 through 32767.

You can specify the record size 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. 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 LSRPOOLID in the file resource definition.

CVDA values are:

NOTAPPLIC
The file is not eligible to be accessed in RLS mode because:
- It is a remote file, or
- It refers to a BDAM data set.

NOTRLS
The file is to be opened in LSR or NSR mode when it is next opened.

RLS
The file is to 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:

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.
SET FILE

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 could be switched to use a coupling
facility data table at a later date.

UPDATE(cvda)
specifies whether the file is read-only or read/write. CVDA values are:

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 model to be used for a coupling facility data table.
CVDA values are:

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 model, 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 could be switched to use a coupling
facility data table at a later date.

Conditions

FILENAMEFOUND
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 LSRPOOLID is specified for a non-VSAM data set.
11 LSRPOOLID is not in the range 1–6, 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.
SET 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 not valid for a BDAM file (must be VSAM for a data table).
32 The TABLE option is not valid 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, nor a new DSNB connected, for this file because the file has deferred work outstanding, for which there are retained or repeatable read locks. This means there is at least one shunted UOW, 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. This means there is at least one shunted UOW, 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 may 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. This means there is at least one shunted UOW, awaiting completion, that has made changes to this file.
44 A file open request cannot be satisfied because the file references a data set that has been marked as unavailable by a SET DSNANE UNAVAILABLE command.
45 A file open request cannot be satisfied because the file references an RLS-mode data set that has been quiesced by a SET DSNANE QUIESCED command.
46 A file open request cannot be satisfied because the file references 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 DSNANE QUIESCED command.
A file open request cannot be satisfied because the file references a data set for which its ICF Catalog entry indicates that a recovery is pending, or is in progress (for example, a CICSVR job is running).

An invalid CVDA is specified for the READINTEG option.

An attempt has been made to open an RLS file when RLS is not supported, either because the level of VSAM does not support RLS or because RLS=NO has been specified during system initialization.

An invalid CVDA is specified for the RLSACCESS option.

An attempt has been made to specify RLS access for a BDAM data set.

An attempt has been made to specify a CICS-maintained data table for a file defined with RLS access.

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 which 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 which it references.
- the file is being opened in non-RLS mode and this region has unresolved RLS recovery work against the data set which it references.

LOADTYPE has an invalid CVDA value.

UPDATEMODEL has an invalid CVDA value.

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.

CFDTPOOL is not specified for a file that refers to a coupling facility data table.

KEYLENGTH is not specified for a file that refers to a coupling facility data table, and which specifies LOAD=NO.

An invalid KEYLENGTH is specified. The KEYLENGTH must be in the range 1 through 16 for a coupling facility data table.

RECORDSIZE is not specified for a file that refers to a coupling facility data table that specifies LOAD=NO.

An invalid RECORDSIZE is specified. RECORDSIZE must be in the range 0 through 32767 bytes.

OPEN is specified for a file that refers to a coupling facility data table, but OPEN processing has failed because:
- The file attributes do not match those specified when the CFDT was created, or
- A keylength or recordsize has been specified that exceeds the maximum supported.

OPEN is specified for a file that refers to a coupling facility data table, but OPEN processing has failed because the server is not available.

An invalid CFDTPOOL name is specified.

An invalid TABLE name is specified.

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.

The DSNAME is invalid.

IOERR

RESP2 values:
SET FILE

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, the CICS console should be examined for messages providing more information on the error.

Depending on how you are viewing the RESP2 value, you should be aware that it may 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

```cics
EXEC CICS SET FILE ('FILE12')
  WAIT
  CLOSED
  DISABLED
  DELETABLE
  LSRPOOLID(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 has been CLOSED, it is to be DISABLED. The records on FILE12 are then to be marked DELETABLE, LSRPOOL number 7 is to be associated with the file, and up to 50 concurrent operations are to be 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 it is possible for another task to delete the file after the first SET command but before the second SET command.

---

4. Resource management system refers to any of the catalogue 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).
SET HOST

Sets the status of a virtual host to enabled or disabled.

```
SET HOST
  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 IRC

Open or close interregion communication.

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.

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
CICS was initialized without IRC support (ISC=NO).

No connection has been defined.

The VTAM APPLID for this CICS is blanks; IRC requires a non-blank APPLID.

Another CICS using IRC has the same VTAM APPLID as this one; unique names are required.

IRC rejected the open of this CICS because it had already reached the maximum number of logons.

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.

**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.
**SET JOURNALNAME**

**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 DFHJnn, 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 supported in releases of CICS earlier than CICS Transaction Server for z/OS for changing the OPENSTATUS setting of a journal.

Description

For CICS Transaction Server for z/OS, this command is replaced by the SET JOURNALNAME command. All the options on the SET 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.
SET JVMPOOL

Change the status of the JVM pool in the CICS region.

<table>
<thead>
<tr>
<th>Options</th>
<th>JVMPool Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>JVMLVL0TRACE(data-area)</td>
<td>STATUS(cvda)</td>
</tr>
<tr>
<td>JVMLVL1TRACE(data-area)</td>
<td>TERMINATE(cvda)</td>
</tr>
<tr>
<td>JVMLVL2TRACE(data-area)</td>
<td>PHASEOUT</td>
</tr>
<tr>
<td>JVMUSERTRACE(data-area)</td>
<td>PURGE</td>
</tr>
<tr>
<td></td>
<td>FORCEPURGE</td>
</tr>
</tbody>
</table>

Conditions: INVREQ, NOTAUTH

Description

The SET JVMPOOL command allows you to change the status of the pool of JVMs in the CICS region or to terminate the pool. There can only be one pool of JVMs in a CICS region; therefore there is no name or identifier required on this command.

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 system initialization parameter SPCTRSJ. 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:

**ENABLED**

The pool status is set to enabled for use and Java programs can execute using JVMs from the pool.

**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 are allowed to execute to completion.

**TERMINATE(cvda)**

specifies that the JVM pool is to be terminated. All the JVMs in the pool, both worker JVMs using the shared class cache and JVMs running independently of the shared class cache, are terminated. No more JVMs can use the existing shared class cache, and it is deleted when all the worker JVMs that were dependent on it have been terminated. If autostart is enabled for the shared class cache, a new shared class cache is started as soon as a new JVM requests it. The CVDA values are:

**PHASEOUT**

All JVMs in the pool are marked for deletion. The JVMs are actually deleted when they finish running their current Java program. The
SET JVMPOOL

shared class cache is deleted when all the worker JVMs that were
dependent on it have been terminated.

PURGE
All tasks using JVMs in the pool are terminated using the SET TASK
PURGE mechanism, and the JVMs are terminated. The shared class
cache is deleted when all the worker JVMs that were dependent on it
have been terminated.

FORCEPURGE
All tasks using JVMs in the pool are terminated by the SET TASK
FORCEPURGE mechanism, and the JVMs are terminated. The shared
class cache is deleted when all the worker JVMs that were dependent
on it have been terminated.

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 MODENAME

Change the number of sessions in an APPC session group.

**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 an implicit syncpoint).

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
SET MODENAME

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.

Description

The SET MONITOR command allows you 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). Monitor 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 two types of data for each individual task:
- Performance data (types and counts of CICS commands, timings, and so on)
- 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).

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; 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.

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
SET MONITOR

Exception class data is written at the end of the event to which the exception corresponds. Performance class and transaction resource data is written only 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 syncpoint, if the SYNCPOINTST value is SYNCPOINT
- When a user event monitoring point with the DELIVER option occurs.

If you change STATUS from ON to OFF, CICS stops accumulating and recording monitor 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, you should specify NOPERF in any command that sets monitoring OFF, to ensure that buffers containing recorded data for completed tasks are flushed; some of this data can be lost otherwise.

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 during execution of a task, 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—syncpoint 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 may be combined. Also, if you switch transaction resource monitoring (RESRCECLASS) between RESRCE and NORESRCE, the rules are the same as for performance class.

Options

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.

EXCEPTCLASS (cvda)
specifies whether the exception class of monitoring data is to be recorded when monitoring is active. CVDA values are:

EXCEPT
Exception data is to be recorded.
SET MONITOR

NOEXCEPT
  Exception data is not to be recorded.

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.

  The frequency interval can be expressed in several ways:
  - A 4-byte packed decimal composite, in the format 0hhmmss+, 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 means 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
  (FREQUENCYMIN or FREQUENCYSEC used alone can exceed 59). For
  example, you could 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).

PERFCLASS(cvda)
  specifies whether the performance class of monitoring data is to be recorded
  when monitoring is active. CVDA values are:

    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
  disabled. CVDA values are:

    NORESRCE  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.
SET MONITOR

**STATUS**(cvda)
specifies whether CICS monitoring is to be active or disabled. CVDA values are:

- **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) within tasks that contain multiple UOWs. A UOW within a task ends when a syncpoint 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 syncpoint, the UOW does not end. CVDA values are:

- **NOSYNCPOINT**  Performance data is to be combined over all UOWs in a task.
- **SYNCPOINT**    Performance data is to be recorded separately for each UOW.

**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.

**NOTAUTH**
RESP2 values:

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

Change the tracing of a VTAM terminal.

**Description**

The SET NETNAME command allows you to control CICS VTAM exit tracing for a particular VTAM terminal (or session). You can specify any VTAM 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 VTAM 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 VTAM 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 VTAM terminal.

**NOTAUTH**

RESP2 values:

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

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.
**SET PIPELINE**

Change the status of an installed PIPELINE.

**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.

**Conditions**

- **INVREQ**
  - RESP2 values:
    - 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 (BTS) process-type.

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 BTS in a single CICS region, you can use the SET PROCESSTYPE command to modify your process-types. However, if you are using BTS 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.
SET PROCESSTYPE

For details of the records that are written from the process, activity primary, and activity secondary audit points, see the CICS Business Transaction Services manual.

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.

**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, we use the word *module* to refer to 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 is executed under EDF. CVDA values are:

- **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 appear.

- **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

---

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
SET PROGRAM

searches the concatenated DFHRPL libraries, and returns an IOERR exception if it cannot locate a copy there. CVDA values are:

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 the program is not in use.)

# This option is not valid for JVM programs, and is ignored if specified.
# You can achieve a synchronized replacement of a JVM program by
# using the command:
#    SET JVMPOOL PHASEOUT

PHASEIN
The refresh is to occur whether or not 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.

# This option is permitted for JVM programs, but it is ignored if specified.
# You can achieve a synchronized replacement of a JVM program by
# using the command:
#    SET JVMPOOL PHASEOUT

COPY cannot be specified for any module currently loaded with the HOLD option, or for any program defined as remote.

EXECUTIONSET(cvda) (programs only)
specifies whether the program is to be restricted to executing the distributed program link (DPL) subset of the CICS API. EXECUTIONSET applies only to executable programs, and governs the API only when a program executes locally. (Programs are always restricted to this subset when invoked remotely—that is, when executing at or below the level of a program invoked by DPL.) CVDA values are:

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 invoked remotely.

The EXECUTIONSET attribute applies only:
• to programs which are being linked to, and not to those 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) (JVM 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, you should specify a JVMCLASS value. If you specify NOJVM in the RUNTIME option, any value in the JVMCLASS option is ignored when the program is executed.

JVMPROFILE(8-character data-value) (JVM programs only)
specifies the name of a JVM profile that is to be used for the JVM in which this program runs. Any instances of this program that are currently running in a JVM with the old JVM profile are unaffected, and are allowed to finish running.
SET PROGRAM

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 HFS file name.

**PROGRAM**(*data-value*)
specifies the 8-character name of the program, map set, or partition set definition to be changed.

**RUNTIME**(*cvda*) (JVM programs only)
specifies whether the program is to run under the control of a JVM. CVDA values are:

- **JVM** The program is to run under the control of a JVM. You should specify a JVMCLASS value.
- **NOJVM** The program is not to run under the control of 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 language, and hence whether it will run with LE370, is determined.

**SHARESTATUS**(*cvda*)
specifies where CICS should obtain the module the next time a new copy is required. A new copy request can result from either an explicit request (SET PROGRAM COPY or the CEMT equivalent) or from a command that requires the module that is issued when CICS does not currently have a copy. CVDA values are:

- **PRIVATE** The module is to be loaded from the concatenated libraries named on the DFHRPL 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 JVM programs is ignored.

**STATUS**(*cvda*)
specifies whether the module is to be available for use. CVDA values are:

- **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 invoked 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 invoked through a CICS program link request, but they have no effect if the program is invoked 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.
SET PROGRAM

For this purpose, CICS defines “different” to mean a switch from a copy loaded from the DFHRPL libraries to the link-pack area copy or vice-versa, or a copy loaded from a disk location different from that of the current copy. CVDA values are:

**NEWCOPY**
The new copy is different.

**OLDCOPY**
The new copy is not different. This value is always returned for JVM 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.
24 DEBUG was specified but RUNTIME was not set to JVM.
25 JVMCLASS contains embedded blanks or null (x'00') characters.
26 JVMPROFILE contains invalid characters or embedded blanks.

**IOERR**
RESP2 values:
8  Either the COPY option or the RUNTIME(NOJVM) option was specified but CICS could not 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.
Examples

EXEC CICS SET PROGRAM ('PROGA')
PHASEIN
PRIVATE
DISABLED

In this example, CICS is to make module PROGA unavailable for new requests, and to locate a new copy in one of the DFHRPL libraries. Any copies of PROGA with a non-zero RESCOUNT will remain until no longer in use, but new requests for PROGA will fail until PROGA is set to ENABLED status. On the first request after the module is enabled, CICS loads the new copy and makes it the current one.
SET STATISTICS

Change the recording of CICS statistics.

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,

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
the option that sets the initial value for the RECORDING switch, set to ON.

The CICS Performance Guide contains more detail about CICS statistics, including the values to which various types of statistics are reinitialized.

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 0hhmmss+, 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 0hhmmss+.

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.)
SET STATISTICS

**INTERVAL**(*data-value*)
specifies the recording interval for system statistics, as a 4-byte packed decimal field in the format **Ohhmms**. 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, described on page 466 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 ‘CICS statistics tables’ in 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.
SET STATISTICS

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

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 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 option can be specified on a SET SYSDUMPCODE REMOVE command.
- **RESET**
SET SYSDUMPCODE

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, 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 the z/OS V1R4.0 MVS Diagnosis: Tools and Service Aids manual.)

When SET SYSDUMPCODE ADD is specified, if you do not also specify DAEOPTION, it defaults to NODAE—regardless of the setting of the SIT DAE parameter.

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 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 executing under MVS/ESA 5.1 and the MVS workload manager.
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.
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.
**SET SYSDUMPCODE**

**SHUTDOWN**
The system is to be shut down.

NOSHUTDOWN is assumed if you omit this value from a SET
SYSDUMPCODE ADD command.

**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 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). 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.
- **14**   RELATED requires MVS/ESA 5.1.
- **15**   DAEOPTION 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.
SET SYSDUMPCODE

NOTFND
  RESP2 values:
  1    The dump code cannot be found.
SET SYSTEM

Change CICS system option values.

Notes:
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

The SET SYSTEM command allows you to change the values of some of the options that control the execution of your CICS system.
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:

- **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 Application Programming 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 the 16MB boundary. If DSALIMIT specifies a value lower than the current limit, CICS may not be able to implement the new limit immediately, but will attempt to do so over time as storage is freed. The range for DSALIMIT is 2MB–16MB.

**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:

- **NOSYSDUMP**
  System dumps are to be suppressed.
SET SYSTEM

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 four individual dynamic storage areas that reside above the 16MB boundary. If EDSALIMIT specifies a value lower than the current limit, CICS may not be able to implement the new limit immediately, but will attempt to do so over time as storage is freed. The range for EDSALIMIT is 10MB–2GB.

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:

- 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 OPENAPI programs or C/C++ programs compiled with XPLINK.

- 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. See the CICS System Definition Guide for information about the LOGDEFER parameter and associated SIT parameter LGDFINT.

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.
SET SYSTEM

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 in CICS Transaction Server for z/OS either, although they did in earlier releases. 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:

- **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:

**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 395).

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 PRTYAGING is not in the range 0–65535.
15 SCANDELAY is not in the range 0–5000.
20 DSALIMIT is not in the range 2MB to 16MB.
SET SYSTEM

21 EDSALIMIT is not in the range 10MB to 2GB.
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**

```
SET TASK(data-value)  [PRIORITY(data-value)]  [PURGETYPE(cvda)]
```

**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

**Conditions:** INVREQ, 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.
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.

**Conditions**

**INVREQ**

RESP2 values:

1. PURGETYPE has an invalid CVDA value.
2. PRIORIRITY is not in the range 0-255.
3. The task is not in a valid state for purging. Any one of the following may apply:
   - The target transaction is defined with SPURGE = NO
   - The target transaction is a CICS created task (that is, it is a system task)
   - The target transaction has already been scheduled for deferred purge (that is, a previous PURGE has resulted in Resp2=13)
   - The target transaction is in termination
   - The target transaction is not suspended or waiting.
4. No previous attempt has been made to forcepurge the task.

**NORMAL**

RESP2 values:

13. The task is not in a valid state for purging. The target transaction is either in transactions initialization or in a dispatcher state that does not allow the purge to be actioned immediately. If the target transaction is in transaction initialization it is marked for deferred abend. A flag is set, and at the end of attach processing the transactions will be abended with AKC3. If the target transaction is in a dispatcher state that does allow purge the purge is deferred until either the dispatcher state changes to a state that does allow the purge or deferred abend handler is given control.

**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 protected by CICS and is not eligible for modification with this command.
SET TCLASS

Set the maximum number of tasks in a transaction class.

**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 611 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.

**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.

**NEWMAXSOCKETS**(data-value)

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. NEWMAXSOCKETS 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.

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

**Conditions:** INVREQ, IOERR, NOTAUTH, NOTSUPERUSER
SET TCPIP

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)
5 TCPIP already closed (for SET CLOSED|IMMCLOSE)
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 TCPIPSERVICEs 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 TCPIPSERVICE**
Modify the status of a service using CICS internal TCPIP support.

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.

**Description**
SET TCPIPSERVICE tells CICS to start or stop listening for incoming requests on the port associated with a service using CICS internal TCPIP support, and changes the attributes of the service.

**Note:** This command has no effect on the sockets support provided by the TCPIP for OS/390 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 actually 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 OS/390 WLM macro IWMSRDRS is called and CICS will no longer be a part of the DNS connection optimization. Any other TCPIPSERVICEs that are in the same group (that is, share the same DNSGROUP attribute) are also deregistered.
**SET TCPIPSERVICE**

**OPENSTATUS**(*cvda*)
Changes the status of the service. CVDA values are:

**OPEN**
CICS starts listening for incoming requests on the specified port.

**CLOSED**
CICS stops accepting new connections for this service. The service is closed once all related sockets are closed. When using HTTP persistent connections, the client may choose to keep the connection open.

**IMMCLOSE**
CICS stops accepting new connections for this service. The service is closed immediately and all related sockets are closed. Tasks initiated using this service will receive an error response only when data transmission is attempted over the socket, which may be at task termination.

**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**(*cvda*)
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 may not allow this name to be changed.

**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:

4 TCPIP is not available (TCPIP=NO)
5 TCPIP status is closed
7 Port is in use
8 CICS is not authorized to use this port
9 TCPIPSERVICE 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 TCPIP is inactive.
14 The TCPIPSERVICE has not been opened because the MAXSOCKETS limit has been reached.
17 The TCPIPSERVICE cannot be opened because PROTOCOL(IIO) was specified and the IIOPLISTENER system initialization parameter is set to NO.

**NOTAUTH**
RESP2 values:

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

NOTFND
    RESP2 values:
    3      the named TCPIPSERVICE is not found.
SET TDQUEUE

Change the attributes of a transient data queue.

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

Conditions: INVREQ, IOERR, NOTAUTH, QIDERR, USERIDERR
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 in-doubt failure, the queue cannot be disabled until the in-doubt 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.
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 in-doubt 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:

1. TRIGGERLEVEL was specified for an extrapartition queue.
2. The TRIGGERLEVEL value is not in the range 0–32767.
3. ATITERMID was specified for an extrapartition queue.
4. ATITRANID was specified for an extrapartition queue.
5. ATIFACILITY was specified for an extrapartition queue.
6. ATIFACILITY has an invalid CVDA value.
7. OPENSTATUS has an invalid CVDA value.
8. OPENSTATUS was specified for an extrapartition queue.
9. ENABLESTATUS has an invalid CVDA value.
10. The queue is remote.
11. The queue is indirect.
12. OPENSTATUS was specified, but the JCL DDNAME to which the queue definition points was not found.
13. SET not possible because the queue was not closed.
14. ATITRANID was specified for an extrapartition queue.
15. The ESM interface is not initialized.
16. CICS has received an unknown response from the ESM.
17. The ESM did not respond.
18. Disabled pending condition.
19. SET not possible because the queue was not disabled.
20. SET not possible because the queue is in-doubt.
21. 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 [z/OS V1R4.0 Security Server RACF Security Administrator's Guide](https://www14.software.ibm.com/webapp/wsola/osa/ibmz/9725SA22-7683/)
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:
SET TDQUEUE

1 The queue cannot be found.

USERIDERR
  RESP2 values:
  28 The user named in ATIUSERID is not known to the ESM.
SET TERMINAL

Change some terminal attributes and cancel outstanding AIDs.

SET TERMINAL

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.
SET TERMINAL

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) (VTAM only)**

is retained only for compatibility purposes. You should use TERMSTATUS in new applications (see page "TERMSTATUS" on page 598).

**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 VTAM 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 ALPRTCOPY 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.

CREATESES(cvdn) (VTAM 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(cvdn)
specifies whether CICS is to honor a disconnect request from the terminal.
CVDA values are:

DISCREQ
CICS honors a disconnect request for a VTAM device, and issues a VTAM CLSDST macro instruction to terminate the VTAM 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 VTAM device.

EXITTRACING(cvdn)
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.
SET TERMINAL

**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 VTAM 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:

**NOPRTCOPY**
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 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-VTAM 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 VTAM application program.

RELREQ
CICS is to release the logical unit, if the logical unit is not currently busy running a transaction.
SET TERMINAL

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 VTAM 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\(\text{data-value}\) (TCAM/DCB remote terminals only)
specifies a 1-character hexadecimal value in the TCAM control byte indicating which segment of the message is to be passed between CICS and TCAM. The meanings are:

- **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.

TERMINAL\(\text{data-value}\)
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\(\text{data-value}\) CANCEL command to remove these requests.

TERMPRIORITY\(\text{data-value}\)
specifies, as a fullword binary value, the priority required for the terminal, relative to other terminals, in the range 0–255.

TERMSTATUS\(_{cvda}\) (VTAM 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.
SET TERMINAL

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.

TTSTATUS\((cvda)\)
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 6 on page 600 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 6. The effect of the UCTRAN parameters

<table>
<thead>
<tr>
<th>Profile</th>
<th>Terminal (TYPETERM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCTRAN (YES)</td>
<td>Tranid: Yes Data: Yes Tranid: No Data: Yes Tranid: Yes Data: Yes</td>
</tr>
<tr>
<td>UCTRAN (NO)</td>
<td>Tranid: Yes Data: Yes Tranid: No Data: No Tranid: Yes Data: No</td>
</tr>
</tbody>
</table>

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 VTAM control component of CICS. CVDA values are:

- **NOZCPTRACE**
  VTAM ZCP tracing is not to be carried out.

- **ZCPTRACE**
  VTAM ZCP tracing is to be carried out.

Conditions

<table>
<thead>
<tr>
<th>INVREQ</th>
<th>RESP2 values:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TERMSTATUS or ACQSTATUS was specified for an IRC session or non-VTAM terminal.</td>
</tr>
<tr>
<td>2</td>
<td>TERMSTATUS or ACQSTATUS has an invalid CVDA value.</td>
</tr>
<tr>
<td>4</td>
<td>ATISTATUS has an invalid CVDA value.</td>
</tr>
<tr>
<td>5</td>
<td>ATISTATUS change would result in NOATI and NOTTI.</td>
</tr>
<tr>
<td>6</td>
<td>CREATESESS was specified for non-VTAM terminal.</td>
</tr>
<tr>
<td>7</td>
<td>CREATESESS has an invalid CVDA value.</td>
</tr>
<tr>
<td>9</td>
<td>PAGESTATUS has an invalid CVDA value.</td>
</tr>
<tr>
<td>11</td>
<td>Trying to put the issuing terminal OUTSERVICE.</td>
</tr>
<tr>
<td>13</td>
<td>SERVSTATUS has an invalid CVDA value.</td>
</tr>
<tr>
<td>15</td>
<td>TERMPRIORITY value not in range 0–255.</td>
</tr>
<tr>
<td>17</td>
<td>NOTTI cannot be specified for the issuing terminal.</td>
</tr>
<tr>
<td>18</td>
<td>TTISTATUS has an invalid CVDA value.</td>
</tr>
<tr>
<td>21</td>
<td>PURGETYPE has an invalid CVDA value.</td>
</tr>
<tr>
<td>22</td>
<td>TRACING has an invalid CVDA value.</td>
</tr>
<tr>
<td>24</td>
<td>Invalid option requested for a remote terminal.</td>
</tr>
<tr>
<td>25</td>
<td>ACQUIRED specified, but terminal is not in service.</td>
</tr>
<tr>
<td>26</td>
<td>PURGE specified, but target task has SPURGE=NO on its associated transaction definition.</td>
</tr>
<tr>
<td>27</td>
<td>EXITTRACING has an invalid CVDA value.</td>
</tr>
<tr>
<td>28</td>
<td>ZCPTRACING has an invalid CVDA value.</td>
</tr>
<tr>
<td>29</td>
<td>EXITTRACING or ZCPTRACING specified for a non-VTAM terminal (or VTAM not installed).</td>
</tr>
<tr>
<td>31</td>
<td>This is a remote terminal with no associated surrogate.</td>
</tr>
<tr>
<td>33</td>
<td>SET TERMINAL is not valid for an LU6.2 (APPC) session.</td>
</tr>
<tr>
<td>34</td>
<td>A permanent transaction has been defined for this terminal (TRANSACTION operand in TERMINAL definition).</td>
</tr>
<tr>
<td>35</td>
<td>Attempt made to change TCAM CONTROL on non-TCAM terminal.</td>
</tr>
<tr>
<td>36</td>
<td>Invalid value supplied for TCAM CONTROL.</td>
</tr>
<tr>
<td>37</td>
<td>Preset signon failed, terminal remains OUTSERVICE.</td>
</tr>
<tr>
<td>38</td>
<td>OBFORMATST has an invalid CVDA value.</td>
</tr>
<tr>
<td>39</td>
<td>RELREQST has an invalid CVDA value.</td>
</tr>
</tbody>
</table>
SET TERMINAL

40 DISCREQST has an invalid CVDA value.
41 ALTPRTCPYST has an invalid CVDA value.
42 PRTCPYST 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 VTAM not available for VTAM terminal.
51 PRINTER and ALTPRINTER option specified for a terminal that is not VTAM 3270 or 3270 compatibility mode.
52 PRTCPYST or ALTPRTCPST option specified for a terminal that is not VTAM 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.

NOMAUT
RESP2 values:
100 The user associated with the issuing task is not authorized to use this command.

TERMDERR
RESP2 values:
23 The named terminal cannot be found.
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). The SET TRACEDEST command allows you to specify which destinations are to receive the 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: INTSTATUS and TRTABSZ (for internal tracing), AUXTR and AUXTRSW (auxiliary tracing), and GTFSTATUS (GTF tracing). See the [CICS System Definition Guide](#) for more information about these values.

Options

**AUXSTATUS(cvda)**

specifies whether auxiliary tracing is to occur; that is, whether trace entries are to be written to the active CICS auxiliary trace data set. (See the SWITCHACTION option for more about auxiliary trace data sets.) CVDA values are:

- **AUXPAUSE**
  
  CICS is to stop writing entries, but is to leave 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.
SET TRACEDEST

AUXSTART
Entries are to be written. The data set is to be opened first if currently closed.

AUXSTOP
Entries are not to be written. The data set is to be closed if open. A subsequent AUXSTART request will cause new entries to be written 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 to be sent to the MVS Generalized Tracing Facility (GTF). CVDA values are:

GTFSTART
Entries are to be sent.

GTFSTOP
Entries are not to be 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 is to occur; that is, whether non-exception trace entries are to be recorded in the internal trace table. (Exception entries are always recorded.) CVDA values are:

INTSTART
Entries are to be recorded.

INTSTOP
Entries are not to be recorded.

SWITCHACTION(cvda)
specifies that the auxiliary trace data sets are to be switched.

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 standby.

If there is only one (or none), SWITCH causes an exception condition, because CICS attempts to open a data set that is not defined.

The CVDA value is:

SWITCH
CICS is to perform a switch.

Note: If you request AUXSTATUS and SWITCHACTION in the same command, AUXSTATUS is set first.

SWITCHSTATUS(cvda)
specifies the action CICS is to take when the current active auxiliary trace data set fills. When this occurs, CICS cannot continue auxiliary tracing unless a
SET TRACEDEST

switch or an AUXSTOP-AUXSTART sequence takes place (see the SWITCHACTION and AUXSTATUS options). CVDA values are:

**NOSWITCH**
- CICS is to take no action.

**SWITCHALL**
- CICS is to switch every time the active data set fills.

**SWITCHNEXT**
- CICS is to switch when the current data set is full, but only once; thereafter NOSWITCH is to be 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, obtains a new table of the requested size, and frees the old one. Data that was in the old table is lost.
- The table is allocated in multiples of 4KB, with a minimum size of 16KB. Consequently, the value you specify is increased to the next multiple of 4, and to 16 if you specify less than 16. The maximum size is 1048576.
- Use caution when setting TABLESIZE to a very high value because there must be enough MVS page storage to satisfy both the request and DSA sizes. The system command DISPLAY ASM MVS can be used to display current information about status and utilization of all MVS page data sets.

**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. The NOSWITCH option tells CICS not to switch when the new active data set fills.
SET TRACEFLAG

Change settings of trace flags.

**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) *(VTAM only)*
specifies which invocations of the CICS VTAM exits are to be traced.

Two types of exit activity can be traced: invocations associated with particular
terminals that have been designated for VTAM 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 VTAM is not installed.

**NOTAUTH**
RESP2 values:
- 100  The user associated with the issuing task is not authorized to use this
command.
The SET TRACETYPE command allows you 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 trace levels 1 and 2 on and all others off.

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 does use 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.

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>APPLICATION</td>
<td>Application</td>
</tr>
<tr>
<td>BA</td>
<td>BUSAPPMGR</td>
<td>Business applications manager</td>
</tr>
<tr>
<td>BF*</td>
<td>BRIDGE</td>
<td>Built-in functions</td>
</tr>
<tr>
<td>BM*</td>
<td>BRIDGE</td>
<td>Basic mapping support</td>
</tr>
<tr>
<td>BR*</td>
<td>BRIDGE</td>
<td>3270 Bridge</td>
</tr>
<tr>
<td>CP*</td>
<td>CPI</td>
<td>Common programming interface</td>
</tr>
<tr>
<td>DC*</td>
<td>DUMP</td>
<td>Dump control</td>
</tr>
<tr>
<td>DD</td>
<td>DIRMGR</td>
<td>Directory manager</td>
</tr>
<tr>
<td>DH</td>
<td></td>
<td>Document handling</td>
</tr>
<tr>
<td>DI*</td>
<td>DUMP</td>
<td>Batch data interchange</td>
</tr>
<tr>
<td>DM</td>
<td>DOMAINMGR</td>
<td>Domain manager</td>
</tr>
<tr>
<td>DP</td>
<td>DISPATCHER</td>
<td>Debugging Profiles domain</td>
</tr>
<tr>
<td>DS</td>
<td>DISPATCHER</td>
<td>Dispatch manager</td>
</tr>
<tr>
<td>DU</td>
<td>DUMP</td>
<td>Dump manager</td>
</tr>
<tr>
<td>EI*</td>
<td>EXEC</td>
<td>EXEC interface</td>
</tr>
<tr>
<td>EJ</td>
<td>ENTJAVA</td>
<td>Enterprise Java domain</td>
</tr>
<tr>
<td>EM</td>
<td>EVENTMGR</td>
<td>Event manager</td>
</tr>
<tr>
<td>FC*</td>
<td>GLOBALCATLG</td>
<td>CICS global catalog manager</td>
</tr>
<tr>
<td>GC</td>
<td>GLOBALCATLG</td>
<td>Interval control</td>
</tr>
<tr>
<td>IC*</td>
<td>EVENTMGR</td>
<td>File control and DL/I</td>
</tr>
<tr>
<td>IE</td>
<td></td>
<td>ECI over TCP/IP domain</td>
</tr>
<tr>
<td>II</td>
<td>IIOP</td>
<td>IIOP domain</td>
</tr>
<tr>
<td>IS*</td>
<td>MESSAGE</td>
<td>Intersystem communication</td>
</tr>
<tr>
<td>KC*</td>
<td>MESSAGE</td>
<td>Task control</td>
</tr>
<tr>
<td>KE</td>
<td>KERNEL</td>
<td>Kernel</td>
</tr>
<tr>
<td>LC</td>
<td>LOCALCATLG</td>
<td>CICS local catalog manager</td>
</tr>
<tr>
<td>LD</td>
<td>LOADER</td>
<td>Program load manager</td>
</tr>
<tr>
<td>LG</td>
<td>LOGMGR</td>
<td>Log manager</td>
</tr>
<tr>
<td>LM</td>
<td>LOCKMGR</td>
<td>Lock manager</td>
</tr>
<tr>
<td>ME</td>
<td>MESSAGE</td>
<td>Message manager</td>
</tr>
<tr>
<td>MN</td>
<td>MONITOR</td>
<td>Monitoring manager</td>
</tr>
<tr>
<td>NQ</td>
<td></td>
<td>Enqueue domain</td>
</tr>
<tr>
<td>OT</td>
<td>OBJECTTRAN</td>
<td>Object Transaction Service (OTS) domain</td>
</tr>
<tr>
<td>PA</td>
<td>PARAMGR</td>
<td>Parameter manager</td>
</tr>
<tr>
<td>PC*</td>
<td>PARAMGR</td>
<td>Program control</td>
</tr>
<tr>
<td>PG</td>
<td>PROGMGR</td>
<td>Program manager</td>
</tr>
<tr>
<td>PI</td>
<td>PipeLine manager domain</td>
<td></td>
</tr>
<tr>
<td>PT</td>
<td>PARTNER</td>
<td>Partner manager</td>
</tr>
<tr>
<td>RI*</td>
<td></td>
<td>Resource manager interface (RMI)</td>
</tr>
<tr>
<td>RM</td>
<td></td>
<td>Recovery manager</td>
</tr>
<tr>
<td>RX</td>
<td>RRS</td>
<td>Resource recovery services</td>
</tr>
<tr>
<td>RZ</td>
<td>REQUESTSTRM</td>
<td>Request streams domain</td>
</tr>
<tr>
<td>SC*</td>
<td></td>
<td>Storage control</td>
</tr>
<tr>
<td>SH</td>
<td></td>
<td>Scheduler services domain for BTS</td>
</tr>
<tr>
<td>SJ</td>
<td>SJVM</td>
<td>CICS JVM domain</td>
</tr>
<tr>
<td>SM</td>
<td>STORAGE</td>
<td>Storage manager</td>
</tr>
<tr>
<td>SO</td>
<td>SOCKETS</td>
<td>Sockets</td>
</tr>
<tr>
<td>ST</td>
<td>STATISTICS</td>
<td>Statistics manager</td>
</tr>
<tr>
<td>SZ*</td>
<td>STATISTICS</td>
<td>Front-end programming interface</td>
</tr>
<tr>
<td>TC*</td>
<td></td>
<td>Terminal control</td>
</tr>
<tr>
<td>TD*</td>
<td></td>
<td>Transient data</td>
</tr>
<tr>
<td>TI</td>
<td>TIMER</td>
<td>Timer manager</td>
</tr>
</tbody>
</table>
SET TRACETYPE

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>TR</td>
<td>TRACE</td>
<td>Trace manager</td>
</tr>
<tr>
<td>TS</td>
<td>Temporary storage</td>
<td></td>
</tr>
<tr>
<td>UE*</td>
<td>User exit interface</td>
<td></td>
</tr>
<tr>
<td>US</td>
<td>USER</td>
<td>User interface</td>
</tr>
<tr>
<td>WB</td>
<td>WEB</td>
<td>Web interface</td>
</tr>
<tr>
<td>XM</td>
<td>TRANMGR</td>
<td>Transaction manager</td>
</tr>
<tr>
<td>XS</td>
<td>SECURITY</td>
<td>Security manager</td>
</tr>
</tbody>
</table>

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.

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

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.

**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.
SET TRANCLASS

(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 DFHTCLnn, where nn 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.

**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** An entry for this code is to be removed from the table.
- **RESET** The maximum count is to be set to the default.

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
SET TRANDUMPCODE

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.

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:

NOSYSDUMP
A system dump is not to be taken.

SYSDUMP
A system dump is to be taken.
SET TRANDUMPCODE

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.
SET TRANUTDOWNCODE

NOTFND
  RESP2 values:
    1   The dump code cannot be found.
SET TRANSACTION

Change a TRANSACTION definition.

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 to abend dumps and has no effect on DUMP TRANSACTION commands.
SET TRANSACTION

**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 prior to 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. 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.
SET TRANSACTION

**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. When executed under CICS Transaction Server for z/OS, 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 DFHTCLnn, 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.
SET TRANSACTION

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.

This section applies also to the alternative command SET TSQNAME. Use either to delete a queue with a name up to 8 characters long, use SET TSQNAME to delete a queue with a name up to 16 characters long.

### 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(data-value)**
      - if this option is specified, the queue is only deleted if its last used interval is greater or equal to the value specified.

- **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 values:
  - RESP2 values:
SET TSQUEUE / TSQNAME

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 filetype 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

<table>
<thead>
<tr>
<th>SET UOW(data-value)</th>
<th>UOWSTATE(cvda)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>COMMIT</td>
</tr>
<tr>
<td></td>
<td>BACKOUT</td>
</tr>
<tr>
<td></td>
<td>FORCE</td>
</tr>
</tbody>
</table>

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 transaction’s in-doubt period.

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:

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 in-doubt. 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:
SET UOW

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(data-value) ACTION(cvda)
  DELETE
```

Conditions: INVREQ, NOTAUTH, UOWLNOKFOUND

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 UOW-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 in-doubt, 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.
SET UOWLINK

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

ENABLESTATUS(cvda)

LOCATION(data-area)

REDIRECTTYPE(cvda)

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.

**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:
**SET UOWLINK**

**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

This command is supported in releases of CICS earlier than CICS Transaction Server for OS/390 to add, remove, or change availability of a named journal volume.

Description

SET 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.

Conditions

VOLIDERR

RESP2 values:

1  The program has issued a SET VOLUME command. This command is withdrawn.
SET VTAM

Modify the CICS VTAM connection.

Description

The SET VTAM command allows you to:

- Establish or terminate the CICS connection to VTAM
- Modify the persistent session delay interval value that CICS passes to VTAM
- Deregister CICS from membership of a VTAM generic resource.

Options

DEREGISTERED

specifies that CICS is to be removed from the VTAM generic resource that it is currently a member of. If you deregister a region from membership of a generic resource, you should end any affinities that it owns—see the PERFORM ENDAFFINITY command.

Generic resources are described in the CICS Intercommunication Guide

OPENSTATUS(cvda)

specifies whether or not CICS is to have a connection to VTAM (that is, whether the VTAM ACB is to be open or closed) and, if CICS must close the ACB to comply, how the shutdown should be done. CVDA values are:

CLOSED

The connection is to be closed. If it is currently open, CICS is to quiesce all VTAM activity and then close the VTAM ACB. Tasks using VTAM terminals or sessions are allowed to complete before closure, but new tasks requiring VTAM are not begun.

FORCECLOSE

The connection is to be closed. If currently open, CICS is to close the VTAM ACB immediately. Both VTAM sessions and tasks using VTAM terminate abnormally as a result.

IMMCLOSE

The connection is to be closed. If currently open, CICS is to terminate

For more information about the use of CVDAs, see "CICS-value data areas (CVDAs)" on page 8.
abnormally any tasks using VTAM immediately, do an orderly shutdown of all its VTAM sessions, and then close the VTAM ACB.

**OPEN**  A connection is to be open. If the VTAM ACB is closed, CICS is to open it.

If CICS is using VTAM multi-node persistent sessions, and VTAM has been restarted after an abend, opening the VTAM ACB causes CICS to restore the persistent sessions that VTAM has retained. However, CICS does not restore APPC synclevel 2 sessions, which are unbound.

**PSDINTERVAL**(data-value)

specifies the persistent session delay (PSD) interval value, which determines whether and for how long VTAM is to hold sessions in recovery-pending state after a CICS failure. The range for the value is 0-23:59:59. Zero causes the persistent session feature not to be used; sessions are terminated at the time of the failure.

**Note:** Zero is the only value allowed in a system which is eligible for the extended recovery facility; see the XRFSTATUS option in the INQUIRE SYSTEM command.

When you specify a PSD interval, CICS sets the system initialization option PSDINT (see the CICS System Definition Guide for more about this option). CICS passes this value to VTAM whenever it opens the ACB. This occurs immediately if you specify an OPENSTATUS value of OPEN in the same SET SYSTEM command, or if the VTAM 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 VTAM is below the level that supports persistent sessions (Version 3.4.1), VTAM rejects the PSD request. CICS then sets the PSDINT system option value to zero and returns an INVREQ condition, but goes on to continue any other processing you requested. The INVREQ occurs when the value is passed to VTAM, which may be later than the command that set it, as explained above. Consequently, you can get this condition on a command that does not specify a PSD interval.

The PSD interval can be expressed in several ways:
- A 4-byte packed decimal composite, in the format $0hhmmss+$, 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 could 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 PSD interval, in fullword binary form (see the PSDINTERVAL option).
SET VTAM

**PSDINTMINS**(data-value)

specifies the minutes component of the PSD interval, in fullword binary form (see the PSDINTERVAL option).

**PSDINTSECS**(data-value)

specifies the seconds component of the PSD interval, in fullword binary form (see the PSDINTERVAL option).

**Conditions**

**INVREQ**

<table>
<thead>
<tr>
<th>RESP2 values:</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VTAM is not present in the system.</td>
</tr>
<tr>
<td>2</td>
<td>OPENSTATUS has an invalid CVDA value.</td>
</tr>
<tr>
<td>4</td>
<td>The PSDINTERVAL value is out of range.</td>
</tr>
<tr>
<td>5</td>
<td>The PSDINTHRS value is out of range.</td>
</tr>
<tr>
<td>6</td>
<td>The PSDINTMINS value is out of range.</td>
</tr>
<tr>
<td>7</td>
<td>The PSDINTSECS value is out of range.</td>
</tr>
<tr>
<td>8</td>
<td>A PSDINTERVAL value &gt; 0 was specified in an XRF-eligible system.</td>
</tr>
<tr>
<td>9</td>
<td>VTAM reported an error while an attempt was being made to set the PSD interval.</td>
</tr>
<tr>
<td>10</td>
<td>A PSD interval has been specified but either the VTAM currently in use (or the VTAM library used when the terminal control table was assembled) does not support persistent sessions. The interval may have been specified earlier than this command; see the PSDINTERVAL description. If OPEN was also requested, CICS has opened the VTAM ACB.</td>
</tr>
<tr>
<td>11</td>
<td>The ACB has opened successfully, but an error occurred in at least one of the sessions that persisted from the previous failure.</td>
</tr>
<tr>
<td>12</td>
<td>Your OPEN request did not complete because another task subsequently requested a close of the VTAM connection.</td>
</tr>
<tr>
<td>13</td>
<td>An error occurred during recovery of sessions, and the VTAM ACB is closed as a result.</td>
</tr>
<tr>
<td>14</td>
<td>CICS is performing cleanup processing following a predatory XRF takeover. CICS rejects OPEN requests with this error, without invoking VTAM, during this activity. OPEN requests are processed as usual as soon as cleanup is complete.</td>
</tr>
<tr>
<td>16</td>
<td>Your attempt to deregister CICS from a VTAM generic resource failed because CICS is not registered as a member of a generic resource group.</td>
</tr>
</tbody>
</table>

**IOERR**

<table>
<thead>
<tr>
<th>RESP2 values:</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>An error occurred during the opening of the ACB. If CICS could not process the request, the RESP2 value is 3. If VTAM detected the failure, CICS returns the VTAM FDBK2 code in RESP2: you can look up these errors in the <em>VTAM Programming</em> manual, under ACB OPEN and CLOSE return codes.</td>
</tr>
</tbody>
</table>

**NOTAUTH**

<table>
<thead>
<tr>
<th>RESP2 values:</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>The user associated with the issuing task is not authorized to use this command.</td>
</tr>
</tbody>
</table>
SET WEB

Modify CICS Web support.

**SET WEB**

```
SET WEB
  GARBAGEINT(data-value)  TIMEOUTINT(data-value)
```

**Conditions:** INVREQ, IOERR, NOTAUTH

#

This command is threadsafe.

For more information about the use of CVDAs, see “CICS-value data areas (CVDAs)” on page 8.

**Description**

The SET WEB command allows you to:
- Change Web garbage collection settings.
- Change Web 3270 terminal timeout settings.

**Options**

**GARBAGEINT**(data-value)

specifies, as a fullword, 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. The permitted range of values is 1 to 6000.

**TIMEOUTINT**(data-value)

specifies, as a fullword, the period of time, in minutes, after which inactive Web 3270 sessions are eligible for garbage collection. The permitted range of values is 1 to 60.

**Conditions**

**INVREQ**

RESP2 values are:

11 an invalid value has been supplied for GARBAGEINT or TIMEOUTINT.

**NOTAUTH**

RESP2 values are:

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

SET WEBSERVICE

Use the SET WEBSERVICE command to change the status of an installed WEBSERVICE.

```
SET WEBSERVICE

SET WEBSERVICE(name)

VALIDATIONST(cvda)
```

Conditions: INVREQ, NOTAUTH, NOTFND

Options

**WEBSERVICE**(name)

Specifies the name of the WEBSERVICE.

**VALIDATIONST**(cvda)

Specifies whether full validation is enabled for the WEBSERVICE or not. CVDA values are:

- **VALIDATION**
  Full validation is enabled.

- **NOVALIDATION**
  Full validation is not enabled.

Conditions

**INVREQ**

RESP2 values are:

- 9  VALIDATIONST cannot be changed because the WEBSERVICE is not INSERVICE.

**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 WEBSERVICE cannot be found.
SET WORKREQUEST

To Purge or Forcepurge a specific local task.

**Description**

The SET WORKREQUEST command allows you to purge a specific local IIOP task (terminate it abnormally) using the PURGE or FORCEPURGE option. These actions are not permitted for request receivers, any such attempts fail.

SET WORKREQUEST is only available for IIOP tasks.

Work Requests are identified by tokens. Each token is unique in the local system. It is expressed as an eight character field, where the characters are hexadecimal digits.

For additional background information about the SPI commands that handle Work Requests, see "INQUIRE WORKREQUEST" on page 440.

**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.
SET WORKREQUEST

**Note:** 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.

**Note:** 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.

**Conditions**

**INVREQ**
RESP2 values:
- **2** The attempted purge or forcepurge action is not valid.
- **5** The attempt to purge or forcepurge a local task failed.
- **6** The attempt to purge or forcepurge 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 forcepurge 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.
Appendix A. CICS-value data areas used by all commands

This lists the CICS-value data-area (CVDA) values and their numeric equivalents for all of the EXEC CICS commands. CVDAs are described beginning on page 8.

This appendix consists of three tables:
- one, in alphabetic sequence of the CVDA values;
- one, (beginning on page 652) in numeric sequence of the CVDA values;
- and one, (beginning on page 668) gives the CVDA values returned by the INQUIRE TERMINAL|NETNAME DEVICE command.

### CVDAs and numeric values in alphabetic sequence

<table>
<thead>
<tr>
<th>CVDA</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABEND</td>
<td>900</td>
</tr>
<tr>
<td>ACQFAIL</td>
<td>515</td>
</tr>
<tr>
<td>ACQUIRED</td>
<td>69</td>
</tr>
<tr>
<td>ACQUIRING</td>
<td>71</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>181</td>
</tr>
<tr>
<td>ACTIVITY</td>
<td>1002</td>
</tr>
<tr>
<td>ADD</td>
<td>291</td>
</tr>
<tr>
<td>ADDABLE</td>
<td>41</td>
</tr>
<tr>
<td>ADDFAIL</td>
<td>519</td>
</tr>
<tr>
<td>ADVANCE</td>
<td>265</td>
</tr>
<tr>
<td>AINIT</td>
<td>282</td>
</tr>
<tr>
<td>ALARM</td>
<td>501</td>
</tr>
<tr>
<td>ALLCONN</td>
<td>169</td>
</tr>
<tr>
<td>ALLOCATED</td>
<td>81</td>
</tr>
<tr>
<td>ALLQUERY</td>
<td>431</td>
</tr>
<tr>
<td>ALTERABLE</td>
<td>52</td>
</tr>
<tr>
<td>ALTERNATE</td>
<td>197</td>
</tr>
<tr>
<td>ALTPRTCOPY</td>
<td>446</td>
</tr>
<tr>
<td>ANALYZER</td>
<td>1126</td>
</tr>
<tr>
<td>AND</td>
<td>1005</td>
</tr>
<tr>
<td>ANY</td>
<td>158</td>
</tr>
<tr>
<td>APLKYBD</td>
<td>391</td>
</tr>
<tr>
<td>APLTEXT</td>
<td>393</td>
</tr>
<tr>
<td>APPC</td>
<td>124</td>
</tr>
<tr>
<td>APPCPARALLEL</td>
<td>374</td>
</tr>
<tr>
<td>APPCSINGLE</td>
<td>373</td>
</tr>
<tr>
<td>APPEND</td>
<td>1036</td>
</tr>
<tr>
<td>APPLICATION</td>
<td>559</td>
</tr>
<tr>
<td>APPLNAME</td>
<td>338</td>
</tr>
<tr>
<td>ASACTL</td>
<td>224</td>
</tr>
<tr>
<td>ASCII7</td>
<td>616</td>
</tr>
<tr>
<td>ASCII8</td>
<td>617</td>
</tr>
<tr>
<td>ASSEMBLER</td>
<td>150</td>
</tr>
<tr>
<td>ASSERTED</td>
<td>1104</td>
</tr>
<tr>
<td>ATI</td>
<td>75</td>
</tr>
<tr>
<td>ATTENTION</td>
<td>524</td>
</tr>
<tr>
<td>AUDALARM</td>
<td>395</td>
</tr>
<tr>
<td>AUTOACTIVE</td>
<td>630</td>
</tr>
<tr>
<td>AUTOARCH</td>
<td>262</td>
</tr>
<tr>
<td>AUTOAUTH</td>
<td>1095</td>
</tr>
<tr>
<td>CVDA</td>
<td>Value</td>
</tr>
<tr>
<td>--------------</td>
<td>-------</td>
</tr>
<tr>
<td>AUTOCONN</td>
<td>170</td>
</tr>
<tr>
<td>AUTOINACTIVE</td>
<td>631</td>
</tr>
<tr>
<td>AUTOINIT</td>
<td>284</td>
</tr>
<tr>
<td>AUTOINSTALL</td>
<td>1140</td>
</tr>
<tr>
<td>AUTOPAGEABLE</td>
<td>80</td>
</tr>
<tr>
<td>AUTOREGISTER</td>
<td>1094</td>
</tr>
<tr>
<td>AUTOSTART</td>
<td>618</td>
</tr>
<tr>
<td>AUTOTERMID</td>
<td>1120</td>
</tr>
<tr>
<td>AUXILIARY</td>
<td>247</td>
</tr>
<tr>
<td>AUXPAUSE</td>
<td>313</td>
</tr>
<tr>
<td>AUXSTART</td>
<td>312</td>
</tr>
<tr>
<td>AUXSTOP</td>
<td>314</td>
</tr>
<tr>
<td>AVAILABLE</td>
<td>95</td>
</tr>
<tr>
<td>BACKOUT</td>
<td>192</td>
</tr>
<tr>
<td>BACKTRANS</td>
<td>397</td>
</tr>
<tr>
<td>BACKUNONBWO</td>
<td>800</td>
</tr>
<tr>
<td>BASE</td>
<td>10</td>
</tr>
<tr>
<td>BASEAPI</td>
<td>1052</td>
</tr>
<tr>
<td>BASESPACE</td>
<td>664</td>
</tr>
<tr>
<td>BASICAUTH</td>
<td>1092</td>
</tr>
<tr>
<td>BDAM</td>
<td>2</td>
</tr>
<tr>
<td>BEGINSESSION</td>
<td>510</td>
</tr>
<tr>
<td>BELOW</td>
<td>159</td>
</tr>
<tr>
<td>BGAM</td>
<td>63</td>
</tr>
<tr>
<td>BINARY</td>
<td>1038</td>
</tr>
<tr>
<td>BINIT</td>
<td>283</td>
</tr>
<tr>
<td>BIT</td>
<td>1020</td>
</tr>
<tr>
<td>BLK</td>
<td>47</td>
</tr>
<tr>
<td>BLOCKED</td>
<td>16</td>
</tr>
<tr>
<td>BOTH</td>
<td>1100</td>
</tr>
<tr>
<td>BROWSABLE</td>
<td>39</td>
</tr>
<tr>
<td>BSAM</td>
<td>61</td>
</tr>
<tr>
<td>BUSY</td>
<td>612</td>
</tr>
<tr>
<td>C</td>
<td>149</td>
</tr>
<tr>
<td>CACHE</td>
<td>791</td>
</tr>
<tr>
<td>CANCEL</td>
<td>526</td>
</tr>
<tr>
<td>CANCELLED</td>
<td>624</td>
</tr>
<tr>
<td>CANCELLING</td>
<td>1025</td>
</tr>
<tr>
<td>CD</td>
<td>491</td>
</tr>
<tr>
<td>CEDF</td>
<td>370</td>
</tr>
<tr>
<td>CERTIFICATION</td>
<td>1093</td>
</tr>
<tr>
<td>CFTABLE</td>
<td>833</td>
</tr>
<tr>
<td>CGROUP</td>
<td>902</td>
</tr>
<tr>
<td>CHANNEL</td>
<td>1047</td>
</tr>
<tr>
<td>CHAR</td>
<td>1019</td>
</tr>
<tr>
<td>CHUNKEND</td>
<td>737</td>
</tr>
<tr>
<td>CHUNKNO</td>
<td>735</td>
</tr>
<tr>
<td>CHUNKYES</td>
<td>736</td>
</tr>
<tr>
<td>CICS</td>
<td>660</td>
</tr>
<tr>
<td>CICSAPI</td>
<td>1052</td>
</tr>
<tr>
<td>CICSDATAKEY</td>
<td>379</td>
</tr>
<tr>
<td>CICSECURITY</td>
<td>195</td>
</tr>
<tr>
<td>CICSEXECKEY</td>
<td>381</td>
</tr>
<tr>
<td>CICSTABLE</td>
<td>101</td>
</tr>
<tr>
<td>CVDA</td>
<td>Value</td>
</tr>
<tr>
<td>--------------</td>
<td>-------</td>
</tr>
<tr>
<td>CKOPEN</td>
<td>1055</td>
</tr>
<tr>
<td>CLASSCACHE</td>
<td>1075</td>
</tr>
<tr>
<td>CLEAR</td>
<td>640</td>
</tr>
<tr>
<td>CLICONVERT</td>
<td>743</td>
</tr>
<tr>
<td>CLIENT</td>
<td>1122</td>
</tr>
<tr>
<td>CLIENTAUTH</td>
<td>1032</td>
</tr>
<tr>
<td>CLIENTCERT</td>
<td>1087</td>
</tr>
<tr>
<td>#</td>
<td>741</td>
</tr>
<tr>
<td>CLOSE</td>
<td>19</td>
</tr>
<tr>
<td>CLOSEFAILED</td>
<td>349</td>
</tr>
<tr>
<td>CLOSELEAVE</td>
<td>261</td>
</tr>
<tr>
<td>CLOSEREQUEST</td>
<td>22</td>
</tr>
<tr>
<td>CLOSING</td>
<td>21</td>
</tr>
<tr>
<td>CMDPROT</td>
<td>673</td>
</tr>
<tr>
<td>CMDSECEXT</td>
<td>207</td>
</tr>
<tr>
<td>CMDSECNO</td>
<td>205</td>
</tr>
<tr>
<td>CMDSECYES</td>
<td>207</td>
</tr>
<tr>
<td>COBOL</td>
<td>151</td>
</tr>
<tr>
<td>COBOLII</td>
<td>375</td>
</tr>
<tr>
<td>COLD</td>
<td>788</td>
</tr>
<tr>
<td>COLDACQ</td>
<td>72</td>
</tr>
<tr>
<td>COLDQUERY</td>
<td>433</td>
</tr>
<tr>
<td>COLDSTART</td>
<td>266</td>
</tr>
<tr>
<td>COLOR</td>
<td>399</td>
</tr>
<tr>
<td>COMMAREA</td>
<td>1048</td>
</tr>
<tr>
<td>COMMIT</td>
<td>208</td>
</tr>
<tr>
<td>COMMITFAIL</td>
<td>792</td>
</tr>
<tr>
<td>COMPLETE</td>
<td>1026</td>
</tr>
<tr>
<td>COMPOSITE</td>
<td>1003</td>
</tr>
<tr>
<td>CONFFREE</td>
<td>82</td>
</tr>
<tr>
<td>CONFRECEIVE</td>
<td>83</td>
</tr>
<tr>
<td>CONFSIZEND</td>
<td>84</td>
</tr>
<tr>
<td>CONNECT</td>
<td>903</td>
</tr>
<tr>
<td>CONNECTED</td>
<td>690</td>
</tr>
<tr>
<td>CONNECTING</td>
<td>904</td>
</tr>
<tr>
<td>CONNECTION</td>
<td>755</td>
</tr>
<tr>
<td>CONSISTENT</td>
<td>723</td>
</tr>
<tr>
<td>CONSOLE</td>
<td>66</td>
</tr>
<tr>
<td>CONTENTION</td>
<td>836</td>
</tr>
<tr>
<td>CONTROLSHUT</td>
<td>623</td>
</tr>
<tr>
<td>CONVERSE</td>
<td>600</td>
</tr>
<tr>
<td>CONVIDLE</td>
<td>518</td>
</tr>
<tr>
<td>COORDINATOR</td>
<td>770</td>
</tr>
<tr>
<td>COPID</td>
<td>908</td>
</tr>
<tr>
<td>COPY</td>
<td>401</td>
</tr>
<tr>
<td>CORBA</td>
<td>1101</td>
</tr>
<tr>
<td>CORBASERVER</td>
<td>1141</td>
</tr>
<tr>
<td>CPLUSPLUS</td>
<td>148</td>
</tr>
<tr>
<td>CREATE</td>
<td>67</td>
</tr>
<tr>
<td>CRITICAL</td>
<td>11</td>
</tr>
<tr>
<td>CSIGN</td>
<td>905</td>
</tr>
<tr>
<td>CTERM</td>
<td>906</td>
</tr>
<tr>
<td>CTLGALL</td>
<td>632</td>
</tr>
<tr>
<td>CTLGMODIFY</td>
<td>633</td>
</tr>
<tr>
<td>CVDA</td>
<td>Value</td>
</tr>
<tr>
<td>--------------</td>
<td>-------</td>
</tr>
<tr>
<td>CTLGNONE</td>
<td>634</td>
</tr>
<tr>
<td>CTRLABLE</td>
<td>56</td>
</tr>
<tr>
<td>CTX</td>
<td>907</td>
</tr>
<tr>
<td>CURRENT</td>
<td>260</td>
</tr>
<tr>
<td>CUSERID</td>
<td>909</td>
</tr>
<tr>
<td>DAE</td>
<td>684</td>
</tr>
<tr>
<td>DATA</td>
<td>508</td>
</tr>
<tr>
<td>DATASET</td>
<td>756</td>
</tr>
<tr>
<td>DATASETFULL</td>
<td>793</td>
</tr>
<tr>
<td>DATASTREAM</td>
<td>543</td>
</tr>
<tr>
<td>DB2CONN</td>
<td>1142</td>
</tr>
<tr>
<td>DB2ENTRY</td>
<td>1143</td>
</tr>
<tr>
<td>DEADLOCK</td>
<td>794</td>
</tr>
<tr>
<td>DEBUG</td>
<td>1082</td>
</tr>
<tr>
<td>DEC</td>
<td>46</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>198</td>
</tr>
<tr>
<td>DEFRESP1</td>
<td>497</td>
</tr>
<tr>
<td>DEFRESP1OR2</td>
<td>528</td>
</tr>
<tr>
<td>DEFRESP2</td>
<td>498</td>
</tr>
<tr>
<td>DEFRESP3</td>
<td>499</td>
</tr>
<tr>
<td>DELAY</td>
<td>637</td>
</tr>
<tr>
<td>DELETABLE</td>
<td>43</td>
</tr>
<tr>
<td>DELETE</td>
<td>292</td>
</tr>
<tr>
<td>DELETFAIL</td>
<td>520</td>
</tr>
<tr>
<td>DELEXITERROR</td>
<td>795</td>
</tr>
<tr>
<td>DEREGEROR</td>
<td>679</td>
</tr>
<tr>
<td>DEREGERED</td>
<td>678</td>
</tr>
<tr>
<td>DEST</td>
<td>235</td>
</tr>
<tr>
<td>DISABLED</td>
<td>24</td>
</tr>
<tr>
<td>DISABLEDHOST</td>
<td>1125</td>
</tr>
<tr>
<td>DISABLING</td>
<td>25</td>
</tr>
<tr>
<td>DISCARDFAIL</td>
<td>513</td>
</tr>
<tr>
<td>DISCARDING</td>
<td>1118</td>
</tr>
<tr>
<td>DISCONNING</td>
<td>910</td>
</tr>
<tr>
<td>DISCREQ</td>
<td>444</td>
</tr>
<tr>
<td>DISK1</td>
<td>252</td>
</tr>
<tr>
<td>DISK2</td>
<td>253</td>
</tr>
<tr>
<td>DISK2PAUSE</td>
<td>254</td>
</tr>
<tr>
<td>DISPATCHABLE</td>
<td>228</td>
</tr>
<tr>
<td>DISPATCHER</td>
<td>1144</td>
</tr>
<tr>
<td>DOCTEMPLATE</td>
<td>1145</td>
</tr>
<tr>
<td>DORMANT</td>
<td>1024</td>
</tr>
<tr>
<td>DPLSUBSET</td>
<td>383</td>
</tr>
<tr>
<td>DS3270</td>
<td>615</td>
</tr>
<tr>
<td>DUALCASE</td>
<td>403</td>
</tr>
<tr>
<td>DUMMY</td>
<td>781</td>
</tr>
<tr>
<td>DYNAMIC</td>
<td>178</td>
</tr>
<tr>
<td>EB</td>
<td>490</td>
</tr>
<tr>
<td>EBCDIC</td>
<td>1039</td>
</tr>
<tr>
<td>ECI</td>
<td>1098</td>
</tr>
<tr>
<td>EJB</td>
<td>1102</td>
</tr>
<tr>
<td>EMERGENCY</td>
<td>268</td>
</tr>
<tr>
<td>EMPTY</td>
<td>210</td>
</tr>
<tr>
<td>EMPTYREQ</td>
<td>31</td>
</tr>
<tr>
<td>CVDA</td>
<td>Value</td>
</tr>
<tr>
<td>--------------</td>
<td>-------</td>
</tr>
<tr>
<td>ENABLED</td>
<td>23</td>
</tr>
<tr>
<td>ENABLING</td>
<td>51</td>
</tr>
<tr>
<td>ENDAFFINITY</td>
<td>790</td>
</tr>
<tr>
<td>I</td>
<td></td>
</tr>
<tr>
<td>ENQUEUE</td>
<td>1146</td>
</tr>
<tr>
<td>EQUAL</td>
<td>911</td>
</tr>
<tr>
<td>ESDS</td>
<td>5</td>
</tr>
<tr>
<td>EVENT</td>
<td>334</td>
</tr>
<tr>
<td>EVENTFAIL</td>
<td>1012</td>
</tr>
<tr>
<td>EVENTUAL</td>
<td>3</td>
</tr>
<tr>
<td>EXCEPT</td>
<td>332</td>
</tr>
<tr>
<td>EXCEPTRESP</td>
<td>523</td>
</tr>
<tr>
<td>EXCI</td>
<td>650</td>
</tr>
<tr>
<td>EXCTL</td>
<td>48</td>
</tr>
<tr>
<td>EXCECNQ</td>
<td>751</td>
</tr>
<tr>
<td>EXCECNQADDR</td>
<td>752</td>
</tr>
<tr>
<td>EXIT</td>
<td>912</td>
</tr>
<tr>
<td>EXITTRACE</td>
<td>362</td>
</tr>
<tr>
<td>I</td>
<td></td>
</tr>
<tr>
<td>EXPECT</td>
<td>747</td>
</tr>
<tr>
<td>EXPIRED</td>
<td>1017</td>
</tr>
<tr>
<td>EXTENDEDDS</td>
<td>405</td>
</tr>
<tr>
<td>EXTRA</td>
<td>221</td>
</tr>
<tr>
<td>EXTSECURITY</td>
<td>194</td>
</tr>
<tr>
<td>FAILED</td>
<td>782</td>
</tr>
<tr>
<td>FAILEDBKOUT</td>
<td>357</td>
</tr>
<tr>
<td>FAILINGBKOUT</td>
<td>358</td>
</tr>
<tr>
<td>FCLOSE</td>
<td>273</td>
</tr>
<tr>
<td>FILE</td>
<td>238</td>
</tr>
<tr>
<td>FINALQUIESCE</td>
<td>183</td>
</tr>
<tr>
<td>FINPUT</td>
<td>270</td>
</tr>
<tr>
<td>FIRED</td>
<td>1001</td>
</tr>
<tr>
<td>FIRSTINIT</td>
<td>625</td>
</tr>
<tr>
<td>FIRSTQUIESCE</td>
<td>182</td>
</tr>
<tr>
<td>FIXED</td>
<td>12</td>
</tr>
<tr>
<td>FLUSH</td>
<td>783</td>
</tr>
<tr>
<td>FMH</td>
<td>502</td>
</tr>
<tr>
<td>FMHPARM</td>
<td>385</td>
</tr>
<tr>
<td>FOPEN</td>
<td>272</td>
</tr>
<tr>
<td>FORCE</td>
<td>342</td>
</tr>
<tr>
<td>FORSECANCEL</td>
<td>655</td>
</tr>
<tr>
<td>FORSECLOSE</td>
<td>351</td>
</tr>
<tr>
<td>FORSECLOSING</td>
<td>353</td>
</tr>
<tr>
<td>FORCED</td>
<td>1013</td>
</tr>
<tr>
<td>FORCENURGE</td>
<td>237</td>
</tr>
<tr>
<td>FORCEUOW</td>
<td>342</td>
</tr>
<tr>
<td>FORMATEDF</td>
<td>606</td>
</tr>
<tr>
<td>FORMATTED</td>
<td>542</td>
</tr>
<tr>
<td>FORMFEED</td>
<td>407</td>
</tr>
<tr>
<td>FOUTPUT</td>
<td>271</td>
</tr>
<tr>
<td>FREE</td>
<td>85</td>
</tr>
<tr>
<td>FREEING</td>
<td>94</td>
</tr>
<tr>
<td>FULL</td>
<td>212</td>
</tr>
<tr>
<td>FULLAPI</td>
<td>384</td>
</tr>
<tr>
<td>FULLAUTO</td>
<td>1073</td>
</tr>
<tr>
<td>FWDRECOVABLE</td>
<td>354</td>
</tr>
<tr>
<td>CVDA</td>
<td>Value</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------</td>
</tr>
<tr>
<td>GENERIC</td>
<td>651</td>
</tr>
<tr>
<td>GET</td>
<td>748</td>
</tr>
<tr>
<td>GMT</td>
<td>604</td>
</tr>
<tr>
<td>GOINGOUT</td>
<td>172</td>
</tr>
<tr>
<td>GROUP</td>
<td>913</td>
</tr>
<tr>
<td>GTFSTART</td>
<td>317</td>
</tr>
<tr>
<td>GTFSTOP</td>
<td>318</td>
</tr>
<tr>
<td>HEAD</td>
<td>750</td>
</tr>
<tr>
<td>HEURBACKOUT</td>
<td>758</td>
</tr>
<tr>
<td>HEURCOMMIT</td>
<td>757</td>
</tr>
<tr>
<td>HEX</td>
<td>45</td>
</tr>
<tr>
<td>HFORM</td>
<td>409</td>
</tr>
<tr>
<td>HFSFILE</td>
<td>1040</td>
</tr>
<tr>
<td>HIGH</td>
<td>914</td>
</tr>
<tr>
<td>HILIGHT</td>
<td>413</td>
</tr>
<tr>
<td>HOLD</td>
<td>163</td>
</tr>
<tr>
<td>HOME</td>
<td>1103</td>
</tr>
<tr>
<td>HTTP</td>
<td>1096</td>
</tr>
<tr>
<td>HTTPNO</td>
<td>1034</td>
</tr>
<tr>
<td>HTTPS</td>
<td>1029</td>
</tr>
<tr>
<td>HTTPYES</td>
<td>1033</td>
</tr>
<tr>
<td>IGNORE</td>
<td>1</td>
</tr>
<tr>
<td>IGNORED</td>
<td>1105</td>
</tr>
<tr>
<td>IGNORERR</td>
<td>816</td>
</tr>
<tr>
<td>IIOP</td>
<td>1097</td>
</tr>
<tr>
<td>IMMCLOSE</td>
<td>350</td>
</tr>
<tr>
<td>IMMclosing</td>
<td>352</td>
</tr>
<tr>
<td>IMMEDIATE</td>
<td>2</td>
</tr>
<tr>
<td>IMMQUIESCED</td>
<td>706</td>
</tr>
<tr>
<td>INACTIVE</td>
<td>378</td>
</tr>
<tr>
<td>INBOUND</td>
<td>547</td>
</tr>
<tr>
<td>INCOMPLETE</td>
<td>1014</td>
</tr>
<tr>
<td>INDEXRECFULL</td>
<td>796</td>
</tr>
<tr>
<td>INDIRECT</td>
<td>122</td>
</tr>
<tr>
<td>INDOUBT</td>
<td>620</td>
</tr>
<tr>
<td>INFLIGHT</td>
<td>621</td>
</tr>
<tr>
<td>INITCOMPLETE</td>
<td>628</td>
</tr>
<tr>
<td>INITIAL</td>
<td>789</td>
</tr>
<tr>
<td>INITING</td>
<td>1112</td>
</tr>
<tr>
<td>INOUT</td>
<td>532</td>
</tr>
<tr>
<td>INPUT</td>
<td>226</td>
</tr>
<tr>
<td>INSERVICE</td>
<td>73</td>
</tr>
<tr>
<td>INSTALLED</td>
<td>550</td>
</tr>
<tr>
<td>INSTALLFAIL</td>
<td>512</td>
</tr>
<tr>
<td>INTERNAL</td>
<td>1058</td>
</tr>
<tr>
<td>INTRA</td>
<td>222</td>
</tr>
<tr>
<td>INTSTART</td>
<td>310</td>
</tr>
<tr>
<td>INTSTOP</td>
<td>311</td>
</tr>
<tr>
<td>INVALID</td>
<td>359</td>
</tr>
<tr>
<td>IOERROR</td>
<td>797</td>
</tr>
<tr>
<td>IRC</td>
<td>121</td>
</tr>
<tr>
<td>ISOLATE</td>
<td>658</td>
</tr>
<tr>
<td>JAVA</td>
<td>147</td>
</tr>
<tr>
<td>JOURNALNAME</td>
<td>1147</td>
</tr>
<tr>
<td>CVDA</td>
<td>Value</td>
</tr>
<tr>
<td>--------------</td>
<td>-------</td>
</tr>
<tr>
<td>JOURNALNUM</td>
<td>1148</td>
</tr>
<tr>
<td>JVM</td>
<td>1080</td>
</tr>
<tr>
<td>JVMPOOL</td>
<td>1149</td>
</tr>
<tr>
<td>JVMPROFILE</td>
<td>1150</td>
</tr>
<tr>
<td>JVMPROGRAM</td>
<td>1151</td>
</tr>
<tr>
<td>KATAKANA</td>
<td>415</td>
</tr>
<tr>
<td>KEEP</td>
<td>786</td>
</tr>
<tr>
<td>KEYED</td>
<td>8</td>
</tr>
<tr>
<td>KILL</td>
<td>232</td>
</tr>
<tr>
<td>KSDS</td>
<td>6</td>
</tr>
<tr>
<td>LCKSTRUCFULL</td>
<td>832</td>
</tr>
<tr>
<td>LEAVE</td>
<td>811</td>
</tr>
<tr>
<td>LE370</td>
<td>377</td>
</tr>
<tr>
<td>LIC</td>
<td>493</td>
</tr>
<tr>
<td>LIGHTPEN</td>
<td>417</td>
</tr>
<tr>
<td>LOAD</td>
<td>834</td>
</tr>
<tr>
<td>LOCAL</td>
<td>605</td>
</tr>
<tr>
<td>LOCKING</td>
<td>837</td>
</tr>
<tr>
<td>LOG</td>
<td>54</td>
</tr>
<tr>
<td>LOGICAL</td>
<td>216</td>
</tr>
<tr>
<td>LOGTERM</td>
<td>269</td>
</tr>
<tr>
<td>LOSE</td>
<td>544</td>
</tr>
<tr>
<td>LOW</td>
<td>915</td>
</tr>
<tr>
<td>LPA</td>
<td>165</td>
</tr>
<tr>
<td>LSRPOOL</td>
<td>1152</td>
</tr>
<tr>
<td>LUP</td>
<td>541</td>
</tr>
<tr>
<td>LUSTAT</td>
<td>525</td>
</tr>
<tr>
<td>LU61</td>
<td>125</td>
</tr>
<tr>
<td>MAIN</td>
<td>248</td>
</tr>
<tr>
<td>MAP</td>
<td>155</td>
</tr>
<tr>
<td>MAPSET</td>
<td>155</td>
</tr>
<tr>
<td>MCHCTL</td>
<td>241</td>
</tr>
<tr>
<td>MDT</td>
<td>506</td>
</tr>
<tr>
<td>MOD</td>
<td>813</td>
</tr>
<tr>
<td>MODEANY</td>
<td>189</td>
</tr>
<tr>
<td>MODEL</td>
<td>370</td>
</tr>
<tr>
<td>MODE24</td>
<td>190</td>
</tr>
<tr>
<td>MODE31</td>
<td>191</td>
</tr>
<tr>
<td>MONITOR</td>
<td>1153</td>
</tr>
<tr>
<td>MORE</td>
<td>492</td>
</tr>
<tr>
<td>MSRCONTROL</td>
<td>419</td>
</tr>
<tr>
<td>MVS</td>
<td>780</td>
</tr>
<tr>
<td>MVSTCB</td>
<td>1154</td>
</tr>
<tr>
<td>NEGATIVE</td>
<td>530</td>
</tr>
<tr>
<td>NEW</td>
<td>28</td>
</tr>
<tr>
<td>NEWCOPY</td>
<td>167</td>
</tr>
<tr>
<td>NEWSSESSION</td>
<td>485</td>
</tr>
<tr>
<td>NOALARM</td>
<td>500</td>
</tr>
<tr>
<td>NOALTPRTCOPY</td>
<td>447</td>
</tr>
<tr>
<td>NOANALYZER</td>
<td>1127</td>
</tr>
<tr>
<td>NOAPLKYBD</td>
<td>392</td>
</tr>
<tr>
<td>NOAPLTXT</td>
<td>394</td>
</tr>
<tr>
<td>NOAPPEND</td>
<td>1037</td>
</tr>
<tr>
<td>NOAPPLNAME</td>
<td>339</td>
</tr>
<tr>
<td>CVDA</td>
<td>Value</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------</td>
</tr>
<tr>
<td>NOATI</td>
<td>76</td>
</tr>
<tr>
<td>NOAUDALARM</td>
<td>396</td>
</tr>
<tr>
<td>NOAUTHENTIC</td>
<td>1091</td>
</tr>
<tr>
<td>NOAUTO</td>
<td>1071</td>
</tr>
<tr>
<td>NOAUTOARCH</td>
<td>263</td>
</tr>
<tr>
<td>NOBACKOUT</td>
<td>193</td>
</tr>
<tr>
<td>NOBACKTRANS</td>
<td>398</td>
</tr>
<tr>
<td>NOCEDF</td>
<td>371</td>
</tr>
<tr>
<td>NOCLASSCACHE</td>
<td>1076</td>
</tr>
<tr>
<td>NOCLEAR</td>
<td>641</td>
</tr>
<tr>
<td>NOCOMPILECONVERT</td>
<td>744</td>
</tr>
<tr>
<td>NOCLOSE</td>
<td>742</td>
</tr>
<tr>
<td>NOCMDPROT</td>
<td>674</td>
</tr>
<tr>
<td>NOCOLOR</td>
<td>400</td>
</tr>
<tr>
<td>NOCONNECT</td>
<td>916</td>
</tr>
<tr>
<td>NOCONV</td>
<td>556</td>
</tr>
<tr>
<td>NOCONVERSE</td>
<td>601</td>
</tr>
<tr>
<td>NOCOPY</td>
<td>402</td>
</tr>
<tr>
<td>NOCREATE</td>
<td>68</td>
</tr>
<tr>
<td>NOCTL</td>
<td>223</td>
</tr>
<tr>
<td>NODAE</td>
<td>685</td>
</tr>
<tr>
<td>NODE</td>
<td>1155</td>
</tr>
<tr>
<td>NODEBUG</td>
<td>1083</td>
</tr>
<tr>
<td>NODISCREQ</td>
<td>445</td>
</tr>
<tr>
<td>NODUALCASE</td>
<td>404</td>
</tr>
<tr>
<td>NOEMPTYREQ</td>
<td>32</td>
</tr>
<tr>
<td>NOEVENT</td>
<td>335</td>
</tr>
<tr>
<td>NOEXCEPT</td>
<td>333</td>
</tr>
<tr>
<td>NOEXCTL</td>
<td>49</td>
</tr>
<tr>
<td>NOEXIT</td>
<td>917</td>
</tr>
<tr>
<td>NOEXITTRACE</td>
<td>363</td>
</tr>
<tr>
<td>NOEXTSNEDDDS</td>
<td>406</td>
</tr>
<tr>
<td>NOFMH</td>
<td>503</td>
</tr>
<tr>
<td>NOFMHPARM</td>
<td>386</td>
</tr>
<tr>
<td>NOFORCE</td>
<td>1054</td>
</tr>
<tr>
<td>NOFORMATEDF</td>
<td>607</td>
</tr>
<tr>
<td>NOFORMFEED</td>
<td>408</td>
</tr>
<tr>
<td>NOFORMFEED</td>
<td>410</td>
</tr>
<tr>
<td>NOHFORM</td>
<td>414</td>
</tr>
<tr>
<td>NOHILIGHT</td>
<td>414</td>
</tr>
<tr>
<td>NOHOLD</td>
<td>164</td>
</tr>
<tr>
<td>NOINCONVERT</td>
<td>745</td>
</tr>
<tr>
<td>NOISOLATE</td>
<td>657</td>
</tr>
<tr>
<td>NOJVVM</td>
<td>1081</td>
</tr>
<tr>
<td>NOKATAKANA</td>
<td>416</td>
</tr>
<tr>
<td>NOLIGHTPEN</td>
<td>418</td>
</tr>
<tr>
<td>NOLOAD</td>
<td>835</td>
</tr>
<tr>
<td>NOLOG</td>
<td>55</td>
</tr>
<tr>
<td>NOLOSTLOCKS</td>
<td>710</td>
</tr>
<tr>
<td>NOMDT</td>
<td>507</td>
</tr>
<tr>
<td>NOMSGJRNL</td>
<td>531</td>
</tr>
<tr>
<td>NOMSRECONTROL</td>
<td>420</td>
</tr>
<tr>
<td>NONAUTOCONN</td>
<td>171</td>
</tr>
<tr>
<td>NONCICS</td>
<td>661</td>
</tr>
<tr>
<td>NONCRITICAL</td>
<td>1090</td>
</tr>
<tr>
<td>CVDA</td>
<td>Value</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>NONE</td>
<td>496</td>
</tr>
<tr>
<td>NONLE370</td>
<td>1084</td>
</tr>
<tr>
<td>NOOBFORMAT</td>
<td>422</td>
</tr>
<tr>
<td>NOOBOPERID</td>
<td>388</td>
</tr>
<tr>
<td>NOOUTCONVERT</td>
<td>746</td>
</tr>
<tr>
<td>NOOUTLINE</td>
<td>424</td>
</tr>
<tr>
<td>NOPARTITIONS</td>
<td>426</td>
</tr>
<tr>
<td>NOPERF</td>
<td>331</td>
</tr>
<tr>
<td>NOPHASEOUT</td>
<td>1110</td>
</tr>
<tr>
<td>NOPRESETSEC</td>
<td>243</td>
</tr>
<tr>
<td>NOPRINTADAPT</td>
<td>428</td>
</tr>
<tr>
<td>NOPROGSYMBOLO</td>
<td>430</td>
</tr>
<tr>
<td>NOPRTCOPY</td>
<td>449</td>
</tr>
<tr>
<td>NOQUERY</td>
<td>432</td>
</tr>
<tr>
<td>NORECOVDATA</td>
<td>700</td>
</tr>
<tr>
<td>NOREENTPROT</td>
<td>681</td>
</tr>
<tr>
<td>NORELEASE</td>
<td>918</td>
</tr>
<tr>
<td>NORELREQ</td>
<td>443</td>
</tr>
<tr>
<td>NORESRCE</td>
<td>337</td>
</tr>
<tr>
<td>NORESYNC</td>
<td>934</td>
</tr>
<tr>
<td>NORETAIRED</td>
<td>711</td>
</tr>
<tr>
<td>NOREUSE</td>
<td>294</td>
</tr>
<tr>
<td>NORMAL</td>
<td>1016</td>
</tr>
<tr>
<td>NORMALBKOUT</td>
<td>356</td>
</tr>
<tr>
<td>NORMALRESP</td>
<td>522</td>
</tr>
<tr>
<td>NORMI</td>
<td>775</td>
</tr>
<tr>
<td>NOROOLBACK</td>
<td>919</td>
</tr>
<tr>
<td>NOSECURITY</td>
<td>196</td>
</tr>
<tr>
<td>NOSHUTDOWN</td>
<td>289</td>
</tr>
<tr>
<td>NOSOSI</td>
<td>435</td>
</tr>
<tr>
<td>NOSPI</td>
<td>694</td>
</tr>
<tr>
<td>NOSRVCONVERT</td>
<td>739</td>
</tr>
<tr>
<td>NOSSSL</td>
<td>1031</td>
</tr>
<tr>
<td>NOSTSN</td>
<td>487</td>
</tr>
<tr>
<td>NOSWITCH</td>
<td>285</td>
</tr>
<tr>
<td>NOSYNCPONTE</td>
<td>603</td>
</tr>
<tr>
<td>NOSYSCONNECT</td>
<td>654</td>
</tr>
<tr>
<td>NOSYSDUMP</td>
<td>185</td>
</tr>
<tr>
<td>NOSYSLOG</td>
<td>784</td>
</tr>
<tr>
<td>NOTADDABLE</td>
<td>42</td>
</tr>
<tr>
<td>NOTALTERABLE</td>
<td>53</td>
</tr>
<tr>
<td>NOTAPPLIC</td>
<td>1</td>
</tr>
<tr>
<td>NOTASKSTART</td>
<td>608</td>
</tr>
<tr>
<td>NOTBROWSABLE</td>
<td>40</td>
</tr>
<tr>
<td>NOTBUSY</td>
<td>613</td>
</tr>
<tr>
<td>NOTCDEB</td>
<td>495</td>
</tr>
<tr>
<td>NOTCONNECTED</td>
<td>691</td>
</tr>
<tr>
<td>NOTCTRLABLE</td>
<td>57</td>
</tr>
<tr>
<td>NOTDEFINED</td>
<td>659</td>
</tr>
<tr>
<td>NOTDELETEABLE</td>
<td>44</td>
</tr>
<tr>
<td>NOTDYNAMIC</td>
<td>1021</td>
</tr>
<tr>
<td>NOTEMPTY</td>
<td>211</td>
</tr>
<tr>
<td>NOTERMINAL</td>
<td>214</td>
</tr>
<tr>
<td>NOTEXTKYBD</td>
<td>437</td>
</tr>
</tbody>
</table>

Appendix A. CICS-value data areas used by all commands 645
<table>
<thead>
<tr>
<th>CVDA</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOTEXTPRINT</td>
<td>439</td>
</tr>
<tr>
<td>NOTFIRED</td>
<td>1000</td>
</tr>
<tr>
<td>NOTFWDRCVBLE</td>
<td>361</td>
</tr>
<tr>
<td>NOTINBOUND</td>
<td>546</td>
</tr>
<tr>
<td>NOTINIT</td>
<td>376</td>
</tr>
<tr>
<td>NOTINSTALLED</td>
<td>551</td>
</tr>
<tr>
<td>NOTKEYED</td>
<td>9</td>
</tr>
<tr>
<td>NOTLPA</td>
<td>166</td>
</tr>
<tr>
<td>NOTPENDING</td>
<td>127</td>
</tr>
<tr>
<td>NOTPURGEABLE</td>
<td>161</td>
</tr>
<tr>
<td>NOTRANDUMP</td>
<td>187</td>
</tr>
<tr>
<td>NOTREADABLE</td>
<td>36</td>
</tr>
<tr>
<td>NOTREADY</td>
<td>259</td>
</tr>
<tr>
<td>NOTRECOVABLE</td>
<td>30</td>
</tr>
<tr>
<td>NOTREQUIRED</td>
<td>667</td>
</tr>
<tr>
<td>NOTRLS</td>
<td>721</td>
</tr>
<tr>
<td>NOTROUTABLE</td>
<td>1022</td>
</tr>
<tr>
<td>NOTSOS</td>
<td>669</td>
</tr>
<tr>
<td>NOTSUPPORTED</td>
<td>15</td>
</tr>
<tr>
<td>NOTSUSPENDED</td>
<td>1027</td>
</tr>
<tr>
<td>NOTTABLE</td>
<td>100</td>
</tr>
<tr>
<td>NOTTI</td>
<td>78</td>
</tr>
<tr>
<td>NOTUPDATABLE</td>
<td>38</td>
</tr>
<tr>
<td>NOTWAIT</td>
<td>920</td>
</tr>
<tr>
<td>NOUCTRAN</td>
<td>451</td>
</tr>
<tr>
<td>NOUSER</td>
<td>1011</td>
</tr>
<tr>
<td>NOVALIDATION</td>
<td>441</td>
</tr>
<tr>
<td>NOVFORM</td>
<td>412</td>
</tr>
<tr>
<td>NOWAIT</td>
<td>341</td>
</tr>
<tr>
<td>NOWRITE</td>
<td>275</td>
</tr>
<tr>
<td>NOZCPTRACE</td>
<td>365</td>
</tr>
<tr>
<td>NRS</td>
<td>774</td>
</tr>
<tr>
<td>N906</td>
<td>931</td>
</tr>
<tr>
<td>N906D</td>
<td>921</td>
</tr>
<tr>
<td>OBFORMAT</td>
<td>421</td>
</tr>
<tr>
<td>OBOPERID</td>
<td>387</td>
</tr>
<tr>
<td>OBTAINING</td>
<td>96</td>
</tr>
<tr>
<td>OFF</td>
<td>200</td>
</tr>
<tr>
<td>OK</td>
<td>274</td>
</tr>
<tr>
<td>OLD</td>
<td>26</td>
</tr>
<tr>
<td>OLDCOPY</td>
<td>162</td>
</tr>
<tr>
<td>OLDSSESSION</td>
<td>486</td>
</tr>
<tr>
<td>ON</td>
<td>201</td>
</tr>
<tr>
<td>OPEN</td>
<td>18</td>
</tr>
<tr>
<td>OPENAPI</td>
<td>1053</td>
</tr>
<tr>
<td>OPENERROR</td>
<td>798</td>
</tr>
<tr>
<td>OPENING</td>
<td>20</td>
</tr>
<tr>
<td>OPENINPUT</td>
<td>256</td>
</tr>
<tr>
<td>OPENOUTPUT</td>
<td>257</td>
</tr>
<tr>
<td>OPID</td>
<td>933</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>759</td>
</tr>
<tr>
<td>OR</td>
<td>1006</td>
</tr>
<tr>
<td>OTS</td>
<td>1107</td>
</tr>
<tr>
<td>OTSCOORD</td>
<td>1108</td>
</tr>
<tr>
<td>CVDA</td>
<td>Value</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------</td>
</tr>
<tr>
<td>OUTLINE</td>
<td>423</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>227</td>
</tr>
<tr>
<td>OUTSERVICE</td>
<td>74</td>
</tr>
<tr>
<td>OWNER</td>
<td>753</td>
</tr>
<tr>
<td>PAGEABLE</td>
<td>79</td>
</tr>
<tr>
<td>PARTITIONS</td>
<td>425</td>
</tr>
<tr>
<td>PARTITIONSET</td>
<td>156</td>
</tr>
<tr>
<td>PATH</td>
<td>11</td>
</tr>
<tr>
<td>PDSMEMBER</td>
<td>1035</td>
</tr>
<tr>
<td>PENDBEGIN</td>
<td>558</td>
</tr>
<tr>
<td>PENDDATA</td>
<td>560</td>
</tr>
<tr>
<td>PENDFREE</td>
<td>86</td>
</tr>
<tr>
<td>PENDING</td>
<td>126</td>
</tr>
<tr>
<td>PENDINIT</td>
<td>1113</td>
</tr>
<tr>
<td>PENDPASS</td>
<td>565</td>
</tr>
<tr>
<td>PENDRECEIVE</td>
<td>87</td>
</tr>
<tr>
<td>PENDRELEASE</td>
<td>562</td>
</tr>
<tr>
<td>PENDRESOLVE</td>
<td>1114</td>
</tr>
<tr>
<td>PENDSTART</td>
<td>561</td>
</tr>
<tr>
<td>PENDSTSN</td>
<td>557</td>
</tr>
<tr>
<td>PENDUNRESOL</td>
<td>564</td>
</tr>
<tr>
<td>PERF</td>
<td>330</td>
</tr>
<tr>
<td>PERMANENT</td>
<td>484</td>
</tr>
<tr>
<td>PIPELINE</td>
<td>1124</td>
</tr>
<tr>
<td>PHASEIN</td>
<td>168</td>
</tr>
<tr>
<td>PHASEOUT</td>
<td>1111</td>
</tr>
<tr>
<td>PHYSICAL</td>
<td>215</td>
</tr>
<tr>
<td>PIPELINE</td>
<td>1124</td>
</tr>
<tr>
<td>PLI</td>
<td>152</td>
</tr>
<tr>
<td>PL1</td>
<td>152</td>
</tr>
<tr>
<td>POOL</td>
<td>922</td>
</tr>
<tr>
<td>POSITIVE</td>
<td>529</td>
</tr>
<tr>
<td>POST</td>
<td>636</td>
</tr>
<tr>
<td>PRESETSEC</td>
<td>242</td>
</tr>
<tr>
<td>PRIMARY</td>
<td>110</td>
</tr>
<tr>
<td>PRINTADAPT</td>
<td>427</td>
</tr>
<tr>
<td>PRIVATE</td>
<td>174</td>
</tr>
<tr>
<td>PROCESS</td>
<td>1010</td>
</tr>
<tr>
<td>PROFILE</td>
<td>157</td>
</tr>
<tr>
<td>PROGAUTO</td>
<td>1072</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>154</td>
</tr>
<tr>
<td>PROGSYMBOL</td>
<td>429</td>
</tr>
<tr>
<td>PROTECTED</td>
<td>504</td>
</tr>
<tr>
<td>PRTCOPY</td>
<td>448</td>
</tr>
<tr>
<td>PUBLISH</td>
<td>1085</td>
</tr>
<tr>
<td>PURGE</td>
<td>236</td>
</tr>
<tr>
<td>PURGEABLE</td>
<td>160</td>
</tr>
<tr>
<td>PUT</td>
<td>749</td>
</tr>
<tr>
<td>QR</td>
<td>1057</td>
</tr>
<tr>
<td>QUASIRENT</td>
<td>1050</td>
</tr>
<tr>
<td>QUEUE</td>
<td>814</td>
</tr>
<tr>
<td>QUIESCED</td>
<td>707</td>
</tr>
<tr>
<td>QUIESCEING</td>
<td>708</td>
</tr>
<tr>
<td>READABLE</td>
<td>35</td>
</tr>
</tbody>
</table>

Appendix A. CICS-value data areas used by all commands  647
<table>
<thead>
<tr>
<th>CVDA</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>READBACK</td>
<td>209</td>
</tr>
<tr>
<td>READONLY</td>
<td>275</td>
</tr>
<tr>
<td>READY</td>
<td>258</td>
</tr>
<tr>
<td>REBUILD</td>
<td>923</td>
</tr>
<tr>
<td>RECEIVE</td>
<td>88</td>
</tr>
<tr>
<td>RECEIVER</td>
<td>1046</td>
</tr>
<tr>
<td>RECONNECT</td>
<td>924</td>
</tr>
<tr>
<td>RECOVDATA</td>
<td>701</td>
</tr>
<tr>
<td>RECOVERABLE</td>
<td>29</td>
</tr>
<tr>
<td>RECOVERED</td>
<td>277</td>
</tr>
<tr>
<td>RECOVERLOCKS</td>
<td>712</td>
</tr>
<tr>
<td>RECOV</td>
<td></td>
</tr>
<tr>
<td>RECOVERY</td>
<td>1156</td>
</tr>
<tr>
<td>REENTPROT</td>
<td>680</td>
</tr>
<tr>
<td>REGERROR</td>
<td>677</td>
</tr>
<tr>
<td>REGISTERED</td>
<td>670</td>
</tr>
<tr>
<td>REJECT</td>
<td>815</td>
</tr>
<tr>
<td>RELATED</td>
<td>675</td>
</tr>
<tr>
<td>RELEASE</td>
<td>563</td>
</tr>
<tr>
<td>RELEASED</td>
<td>70</td>
</tr>
<tr>
<td>RELEASING</td>
<td>549</td>
</tr>
<tr>
<td>RELOAD</td>
<td>1078</td>
</tr>
<tr>
<td>RELOADING</td>
<td>1079</td>
</tr>
<tr>
<td>RELREQ</td>
<td>442</td>
</tr>
<tr>
<td>REMLOSTLOCKS</td>
<td>713</td>
</tr>
<tr>
<td>REMOTE</td>
<td>4</td>
</tr>
<tr>
<td>REMOVE</td>
<td>276</td>
</tr>
<tr>
<td>REMSESSION</td>
<td>740</td>
</tr>
<tr>
<td>REMTABLE</td>
<td>103</td>
</tr>
<tr>
<td>REPEATEABLE</td>
<td>724</td>
</tr>
<tr>
<td>REQUIRES</td>
<td></td>
</tr>
<tr>
<td>REQUESTMODEL</td>
<td>1157</td>
</tr>
<tr>
<td>REQUIRED</td>
<td>666</td>
</tr>
<tr>
<td>REREAD</td>
<td>812</td>
</tr>
<tr>
<td>RESET</td>
<td>290</td>
</tr>
<tr>
<td>RESETLOCKS</td>
<td>714</td>
</tr>
<tr>
<td>RESOLVING</td>
<td>1115</td>
</tr>
<tr>
<td>RESRCE</td>
<td>336</td>
</tr>
<tr>
<td>RESSECEXT</td>
<td>204</td>
</tr>
<tr>
<td>RESSECINT</td>
<td>203</td>
</tr>
<tr>
<td>RESSECCNO</td>
<td>202</td>
</tr>
<tr>
<td>RESSECYES</td>
<td>204</td>
</tr>
<tr>
<td>RESPECTED</td>
<td>1109</td>
</tr>
<tr>
<td>RESYNC</td>
<td>702</td>
</tr>
<tr>
<td>RETAINED</td>
<td>715</td>
</tr>
<tr>
<td>RETRACT</td>
<td>1086</td>
</tr>
<tr>
<td>RETRY</td>
<td>716</td>
</tr>
<tr>
<td>REUSE</td>
<td>293</td>
</tr>
<tr>
<td>REVERTED</td>
<td>264</td>
</tr>
<tr>
<td>REWIND</td>
<td>811</td>
</tr>
<tr>
<td>RFC1123</td>
<td>646</td>
</tr>
<tr>
<td>RLS</td>
<td>720</td>
</tr>
<tr>
<td>RLSACTIVE</td>
<td>730</td>
</tr>
<tr>
<td>RLSGONE</td>
<td>799</td>
</tr>
<tr>
<td>RLSINACTIVE</td>
<td>731</td>
</tr>
<tr>
<td>RLSSERVER</td>
<td>761</td>
</tr>
</tbody>
</table>
### Appendix A. CICS-value data areas used by all commands

<table>
<thead>
<tr>
<th>CVDA</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMI</td>
<td>771</td>
</tr>
<tr>
<td>ROLLBACK</td>
<td>89</td>
</tr>
<tr>
<td>ROUTABLE</td>
<td>1023</td>
</tr>
<tr>
<td>ROUTE</td>
<td>638</td>
</tr>
<tr>
<td>RPG</td>
<td>153</td>
</tr>
<tr>
<td>RRCOMMITSFAIL</td>
<td>830</td>
</tr>
<tr>
<td>RRDS</td>
<td>7</td>
</tr>
<tr>
<td>RRINDOUBT</td>
<td>831</td>
</tr>
<tr>
<td>RRMS</td>
<td>1063</td>
</tr>
<tr>
<td>RTR</td>
<td>527</td>
</tr>
<tr>
<td>RU</td>
<td>494</td>
</tr>
<tr>
<td>RUNNING</td>
<td>229</td>
</tr>
<tr>
<td>SCAN</td>
<td>1088</td>
</tr>
<tr>
<td>SCS</td>
<td>614</td>
</tr>
<tr>
<td>SECONDINIT</td>
<td>626</td>
</tr>
<tr>
<td>SEND</td>
<td>90</td>
</tr>
<tr>
<td>SENDER</td>
<td>1045</td>
</tr>
<tr>
<td>SERVER</td>
<td>1123</td>
</tr>
<tr>
<td>SESSION</td>
<td>372</td>
</tr>
<tr>
<td>SESSIONFAIL</td>
<td>517</td>
</tr>
<tr>
<td>SESSIONLOST</td>
<td>516</td>
</tr>
<tr>
<td>SETFAIL</td>
<td>514</td>
</tr>
<tr>
<td>SHARE</td>
<td>27</td>
</tr>
<tr>
<td>SHARED</td>
<td>173</td>
</tr>
<tr>
<td>SHUNTED</td>
<td>762</td>
</tr>
<tr>
<td>SHUTDISABLED</td>
<td>645</td>
</tr>
<tr>
<td>SHUTDOWN</td>
<td>288</td>
</tr>
<tr>
<td>SHUTENABLED</td>
<td>644</td>
</tr>
<tr>
<td>SIGN</td>
<td>925</td>
</tr>
<tr>
<td>SIGNEDOFF</td>
<td>245</td>
</tr>
<tr>
<td>SIGNEDON</td>
<td>244</td>
</tr>
<tr>
<td>SINGLEOFF</td>
<td>324</td>
</tr>
<tr>
<td>SINGLEON</td>
<td>323</td>
</tr>
<tr>
<td>SKIP</td>
<td>810</td>
</tr>
<tr>
<td>SMF</td>
<td>255</td>
</tr>
<tr>
<td>SNA</td>
<td>60</td>
</tr>
<tr>
<td>SOAP</td>
<td>1044</td>
</tr>
<tr>
<td>SOS</td>
<td>668</td>
</tr>
<tr>
<td>SOSABOVE</td>
<td>683</td>
</tr>
<tr>
<td>SOSBELOW</td>
<td>682</td>
</tr>
<tr>
<td>SOSI</td>
<td>434</td>
</tr>
<tr>
<td>SPECIFIC</td>
<td>652</td>
</tr>
<tr>
<td>SPECTRACE</td>
<td>177</td>
</tr>
<tr>
<td>SPI</td>
<td>693</td>
</tr>
<tr>
<td>SPRSTRACE</td>
<td>175</td>
</tr>
<tr>
<td>SQLCODE</td>
<td>926</td>
</tr>
<tr>
<td>SRVCONVERT</td>
<td>738</td>
</tr>
<tr>
<td>SSL</td>
<td>1030</td>
</tr>
<tr>
<td>STANDBY</td>
<td>629</td>
</tr>
<tr>
<td>STANTRACE</td>
<td>176</td>
</tr>
<tr>
<td>START</td>
<td>635</td>
</tr>
<tr>
<td>STARTED</td>
<td>609</td>
</tr>
<tr>
<td>STARTING</td>
<td>772</td>
</tr>
<tr>
<td>STARTUP</td>
<td>180</td>
</tr>
</tbody>
</table>

---

Appendix A. CICS-value data areas used by all commands
<table>
<thead>
<tr>
<th>CVDA</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATIC</td>
<td>179</td>
</tr>
<tr>
<td>STATS</td>
<td>1158</td>
</tr>
<tr>
<td>STOPPED</td>
<td>610</td>
</tr>
<tr>
<td>STORAGE</td>
<td>1159</td>
</tr>
<tr>
<td>STREAMNAME</td>
<td>1160</td>
</tr>
<tr>
<td>STSN</td>
<td>509</td>
</tr>
<tr>
<td>STSNSET</td>
<td>488</td>
</tr>
<tr>
<td>STSNTEST</td>
<td>489</td>
</tr>
<tr>
<td>SUBORDINATE</td>
<td>773</td>
</tr>
<tr>
<td>SUBPOOL</td>
<td>1161</td>
</tr>
<tr>
<td>SUBSPACE</td>
<td>663</td>
</tr>
<tr>
<td>SUPPORTED</td>
<td>1106</td>
</tr>
<tr>
<td>SURROGATE</td>
<td>371</td>
</tr>
<tr>
<td>SUSPENDED</td>
<td>231</td>
</tr>
<tr>
<td>SWITCH</td>
<td>188</td>
</tr>
<tr>
<td>SWITCHALL</td>
<td>287</td>
</tr>
<tr>
<td>SWITCHING</td>
<td>225</td>
</tr>
<tr>
<td>SWITCHNEXT</td>
<td>286</td>
</tr>
<tr>
<td>SYNCFREE</td>
<td>91</td>
</tr>
<tr>
<td>SYNCPOINT</td>
<td>602</td>
</tr>
<tr>
<td>SYNCRECEIVE</td>
<td>92</td>
</tr>
<tr>
<td>SYNCSEND</td>
<td>93</td>
</tr>
<tr>
<td>SYSCONNECT</td>
<td>653</td>
</tr>
<tr>
<td>SYSDUMP</td>
<td>184</td>
</tr>
<tr>
<td>SYSDUMPICODE</td>
<td>1162</td>
</tr>
<tr>
<td>SYSLOG</td>
<td>785</td>
</tr>
<tr>
<td>SYSPLEX</td>
<td>662</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>643</td>
</tr>
<tr>
<td>SYSTEMOFF</td>
<td>320</td>
</tr>
<tr>
<td>SYSTEMON</td>
<td>319</td>
</tr>
<tr>
<td>TABLEMGR</td>
<td>1163</td>
</tr>
<tr>
<td>TAKEOVER</td>
<td>111</td>
</tr>
<tr>
<td>TAPE1</td>
<td>250</td>
</tr>
<tr>
<td>TAPE2</td>
<td>251</td>
</tr>
<tr>
<td>TASK</td>
<td>233</td>
</tr>
<tr>
<td>TASKSUBPOOL</td>
<td>1164</td>
</tr>
<tr>
<td>TASKSTART</td>
<td>611</td>
</tr>
<tr>
<td>TCAM</td>
<td>64</td>
</tr>
<tr>
<td>TCAMSNA</td>
<td>65</td>
</tr>
<tr>
<td>TCEXITALL</td>
<td>366</td>
</tr>
<tr>
<td>TCEXITALLOFF</td>
<td>369</td>
</tr>
<tr>
<td>TCEXITNONE</td>
<td>368</td>
</tr>
<tr>
<td>TCEXITSYSTEM</td>
<td>367</td>
</tr>
<tr>
<td>TCLASS</td>
<td>239</td>
</tr>
<tr>
<td>TCPIP</td>
<td>1165</td>
</tr>
<tr>
<td>TCPIPSERVICE</td>
<td>1166</td>
</tr>
<tr>
<td>TDQ</td>
<td>767</td>
</tr>
<tr>
<td>TDQUEUE</td>
<td>1167</td>
</tr>
<tr>
<td>TEMPLATENAME</td>
<td>1168</td>
</tr>
<tr>
<td>TEMPORARY</td>
<td>483</td>
</tr>
<tr>
<td>TERM</td>
<td>234</td>
</tr>
<tr>
<td>TERMINAL</td>
<td>213</td>
</tr>
<tr>
<td>TERMINATE</td>
<td>1077</td>
</tr>
<tr>
<td>TEXTKYBD</td>
<td>436</td>
</tr>
<tr>
<td>CVDA</td>
<td>Value</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>TEXTPRINT</td>
<td>438</td>
</tr>
<tr>
<td>THIRDINIT</td>
<td>627</td>
</tr>
<tr>
<td>THREADSAFE</td>
<td>1051</td>
</tr>
<tr>
<td>TIME</td>
<td>240</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>511</td>
</tr>
<tr>
<td>TIMER</td>
<td>1004</td>
</tr>
<tr>
<td>TPOOL</td>
<td>932</td>
</tr>
<tr>
<td>TPS55M2</td>
<td>552</td>
</tr>
<tr>
<td>TPS55M3</td>
<td>553</td>
</tr>
<tr>
<td>TPS55M4</td>
<td>554</td>
</tr>
<tr>
<td>TPS55M5</td>
<td>555</td>
</tr>
<tr>
<td>#</td>
<td>TRACE</td>
</tr>
<tr>
<td>I</td>
<td>TRANCLASS</td>
</tr>
<tr>
<td></td>
<td>TRANDUMP</td>
</tr>
<tr>
<td>I</td>
<td>TRANDUMPCODE</td>
</tr>
<tr>
<td></td>
<td>TRANIDONLY</td>
</tr>
<tr>
<td>I</td>
<td>TRANSACTION</td>
</tr>
<tr>
<td></td>
<td>TSQ</td>
</tr>
<tr>
<td>I</td>
<td>TSQUEUE</td>
</tr>
<tr>
<td></td>
<td>TTI</td>
</tr>
<tr>
<td></td>
<td>TWAIT</td>
</tr>
<tr>
<td></td>
<td>TX</td>
</tr>
<tr>
<td></td>
<td>TXID</td>
</tr>
<tr>
<td></td>
<td>T3278M2</td>
</tr>
<tr>
<td></td>
<td>T3278M3</td>
</tr>
<tr>
<td></td>
<td>T3278M4</td>
</tr>
<tr>
<td></td>
<td>T3278M5</td>
</tr>
<tr>
<td></td>
<td>T3279M2</td>
</tr>
<tr>
<td></td>
<td>T3279M3</td>
</tr>
<tr>
<td></td>
<td>T3279M4</td>
</tr>
<tr>
<td></td>
<td>T3279M5</td>
</tr>
<tr>
<td></td>
<td>UCTRAN</td>
</tr>
<tr>
<td></td>
<td>UKOPEN</td>
</tr>
<tr>
<td></td>
<td>UNATTEMPTED</td>
</tr>
<tr>
<td></td>
<td>UNAVAILABLE</td>
</tr>
<tr>
<td></td>
<td>UNBLOCKED</td>
</tr>
<tr>
<td></td>
<td>UNCOMMITTED</td>
</tr>
<tr>
<td></td>
<td>UNCONNECTED</td>
</tr>
<tr>
<td></td>
<td>UNDEFINED</td>
</tr>
<tr>
<td></td>
<td>UNDETERMINED</td>
</tr>
<tr>
<td></td>
<td>UNENABLED</td>
</tr>
<tr>
<td></td>
<td>UNENABLING</td>
</tr>
<tr>
<td></td>
<td>UNEXPECTED</td>
</tr>
<tr>
<td></td>
<td>UNEXPIRED</td>
</tr>
<tr>
<td></td>
<td>UNKNOWN</td>
</tr>
<tr>
<td></td>
<td>UNPROTECTED</td>
</tr>
<tr>
<td></td>
<td>UNQUIESCED</td>
</tr>
<tr>
<td></td>
<td>UNREGISTERED</td>
</tr>
<tr>
<td></td>
<td>UNRESOLVED</td>
</tr>
<tr>
<td></td>
<td>UNSOLDATA</td>
</tr>
<tr>
<td></td>
<td>UNSUSABLE</td>
</tr>
<tr>
<td></td>
<td>UOW</td>
</tr>
<tr>
<td></td>
<td>UPDATABLE</td>
</tr>
<tr>
<td>I</td>
<td>UPDATING</td>
</tr>
<tr>
<td>CVDA</td>
<td>Value</td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>I</td>
<td></td>
</tr>
<tr>
<td>URIMAP</td>
<td>1173</td>
</tr>
<tr>
<td>I</td>
<td></td>
</tr>
<tr>
<td>URMTERMID</td>
<td>1121</td>
</tr>
<tr>
<td>USER</td>
<td>642</td>
</tr>
<tr>
<td>USERDATAKEY</td>
<td>380</td>
</tr>
<tr>
<td>USEREXECKEY</td>
<td>382</td>
</tr>
<tr>
<td>USERID</td>
<td>930</td>
</tr>
<tr>
<td>USEROFF</td>
<td>322</td>
</tr>
<tr>
<td>USERON</td>
<td>321</td>
</tr>
<tr>
<td>USERTABLE</td>
<td></td>
</tr>
<tr>
<td>VALID</td>
<td>102</td>
</tr>
<tr>
<td>VALIDATION</td>
<td>440</td>
</tr>
<tr>
<td>VARIABLE</td>
<td>13</td>
</tr>
<tr>
<td>VERIFY</td>
<td>1099</td>
</tr>
<tr>
<td>VFORM</td>
<td>411</td>
</tr>
<tr>
<td>VRRDS</td>
<td>732</td>
</tr>
<tr>
<td>VSAM</td>
<td>3</td>
</tr>
<tr>
<td>VTAM</td>
<td>60</td>
</tr>
<tr>
<td>WAIT</td>
<td>340</td>
</tr>
<tr>
<td>WAITCOMMIT</td>
<td>763</td>
</tr>
<tr>
<td>WAITER</td>
<td>754</td>
</tr>
<tr>
<td>WAITFORGET</td>
<td>622</td>
</tr>
<tr>
<td>WAITING</td>
<td>765</td>
</tr>
<tr>
<td>WAITRMI</td>
<td>766</td>
</tr>
<tr>
<td>WARMSTART</td>
<td>267</td>
</tr>
<tr>
<td>I</td>
<td></td>
</tr>
<tr>
<td>WEBSERVICE</td>
<td>1174</td>
</tr>
<tr>
<td>I</td>
<td></td>
</tr>
<tr>
<td>WIN</td>
<td>545</td>
</tr>
<tr>
<td>XCF</td>
<td>665</td>
</tr>
<tr>
<td>XM</td>
<td>123</td>
</tr>
<tr>
<td>XNOTDONE</td>
<td>144</td>
</tr>
<tr>
<td>I</td>
<td></td>
</tr>
<tr>
<td>XPLINK</td>
<td>1068</td>
</tr>
<tr>
<td>XOK</td>
<td>143</td>
</tr>
<tr>
<td>ZCPTRACE</td>
<td>364</td>
</tr>
</tbody>
</table>

### CVDAs and numeric values in numeric sequence

<table>
<thead>
<tr>
<th>Value</th>
<th>CVDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IGNORE</td>
</tr>
<tr>
<td>1</td>
<td>NOTAPPLIC</td>
</tr>
<tr>
<td>2</td>
<td>BDAM</td>
</tr>
<tr>
<td>2</td>
<td>IMMEDIATE</td>
</tr>
<tr>
<td>3</td>
<td>EVENTUAL</td>
</tr>
<tr>
<td>3</td>
<td>VSAM</td>
</tr>
<tr>
<td>4</td>
<td>REMOTE</td>
</tr>
<tr>
<td>5</td>
<td>ESDS</td>
</tr>
<tr>
<td>6</td>
<td>KSDS</td>
</tr>
<tr>
<td>7</td>
<td>RRDS</td>
</tr>
<tr>
<td>8</td>
<td>KEYED</td>
</tr>
<tr>
<td>9</td>
<td>NOTKEYED</td>
</tr>
<tr>
<td>10</td>
<td>BASE</td>
</tr>
<tr>
<td>11</td>
<td>CRITICAL</td>
</tr>
<tr>
<td>11</td>
<td>PATH</td>
</tr>
<tr>
<td>12</td>
<td>FIXED</td>
</tr>
<tr>
<td>13</td>
<td>VARIABLE</td>
</tr>
<tr>
<td>Value</td>
<td>CVDA</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------</td>
</tr>
<tr>
<td>14</td>
<td>UNDEFINED</td>
</tr>
<tr>
<td>15</td>
<td>NOTSUPPORTED</td>
</tr>
<tr>
<td>16</td>
<td>BLOCKED</td>
</tr>
<tr>
<td>17</td>
<td>UNBLOCKED</td>
</tr>
<tr>
<td>18</td>
<td>OPEN</td>
</tr>
<tr>
<td>19</td>
<td>CLOSED</td>
</tr>
<tr>
<td>20</td>
<td>OPENING</td>
</tr>
<tr>
<td>21</td>
<td>CLOSING</td>
</tr>
<tr>
<td>22</td>
<td>CLOSEREQUEST</td>
</tr>
<tr>
<td>23</td>
<td>ENABLED</td>
</tr>
<tr>
<td>24</td>
<td>DISABLED</td>
</tr>
<tr>
<td>25</td>
<td>DISABLING</td>
</tr>
<tr>
<td>26</td>
<td>OLD</td>
</tr>
<tr>
<td>27</td>
<td>SHARE</td>
</tr>
<tr>
<td>28</td>
<td>NEW</td>
</tr>
<tr>
<td>29</td>
<td>RECOVERABLE</td>
</tr>
<tr>
<td>30</td>
<td>NOTRECOVABLE</td>
</tr>
<tr>
<td>31</td>
<td>EMPTYREQ</td>
</tr>
<tr>
<td>32</td>
<td>NOEMPTYREQ</td>
</tr>
<tr>
<td>33</td>
<td>UNENABLED</td>
</tr>
<tr>
<td>34</td>
<td>UNENABLING</td>
</tr>
<tr>
<td>35</td>
<td>READABLE</td>
</tr>
<tr>
<td>36</td>
<td>NOTREADABLE</td>
</tr>
<tr>
<td>37</td>
<td>UPDATABLE</td>
</tr>
<tr>
<td>38</td>
<td>NOTUPDATABLE</td>
</tr>
<tr>
<td>39</td>
<td>BROWSABLE</td>
</tr>
<tr>
<td>40</td>
<td>NOTBROWSABLE</td>
</tr>
<tr>
<td>41</td>
<td>ADDABLE</td>
</tr>
<tr>
<td>42</td>
<td>NOTADDABLE</td>
</tr>
<tr>
<td>43</td>
<td>DELETABLE</td>
</tr>
<tr>
<td>44</td>
<td>NOTDELETABLE</td>
</tr>
<tr>
<td>45</td>
<td>HEX</td>
</tr>
<tr>
<td>46</td>
<td>DEC</td>
</tr>
<tr>
<td>47</td>
<td>BLK</td>
</tr>
<tr>
<td>48</td>
<td>EXCTL</td>
</tr>
<tr>
<td>49</td>
<td>NOEXCTL</td>
</tr>
<tr>
<td>51</td>
<td>ENABLING</td>
</tr>
<tr>
<td>52</td>
<td>ALTERABLE</td>
</tr>
<tr>
<td>53</td>
<td>NOTALTERABLE</td>
</tr>
<tr>
<td>54</td>
<td>LOG</td>
</tr>
<tr>
<td>55</td>
<td>NOLOG</td>
</tr>
<tr>
<td>56</td>
<td>CTRLABLE</td>
</tr>
<tr>
<td>57</td>
<td>NOTCTRLABLE</td>
</tr>
<tr>
<td>60</td>
<td>SNA</td>
</tr>
<tr>
<td>60</td>
<td>VTAM</td>
</tr>
<tr>
<td>61</td>
<td>BSAM</td>
</tr>
<tr>
<td>63</td>
<td>BGAM</td>
</tr>
<tr>
<td>64</td>
<td>TCAM</td>
</tr>
<tr>
<td>65</td>
<td>TCAMSNA</td>
</tr>
<tr>
<td>66</td>
<td>CONSOLE</td>
</tr>
<tr>
<td>67</td>
<td>CREATE</td>
</tr>
<tr>
<td>68</td>
<td>NOCREATE</td>
</tr>
<tr>
<td>69</td>
<td>ACQUIRED</td>
</tr>
<tr>
<td>70</td>
<td>RELEASED</td>
</tr>
<tr>
<td>Value</td>
<td>CVDA</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>71</td>
<td>ACQUiring</td>
</tr>
<tr>
<td>72</td>
<td>COLDACQ</td>
</tr>
<tr>
<td>73</td>
<td>INSERVICE</td>
</tr>
<tr>
<td>74</td>
<td>OUTSERVICE</td>
</tr>
<tr>
<td>75</td>
<td>ATI</td>
</tr>
<tr>
<td>76</td>
<td>NOATI</td>
</tr>
<tr>
<td>77</td>
<td>TTI</td>
</tr>
<tr>
<td>78</td>
<td>NOTTI</td>
</tr>
<tr>
<td>79</td>
<td>PAGEABLE</td>
</tr>
<tr>
<td>80</td>
<td>AUTOPAGEABLE</td>
</tr>
<tr>
<td>81</td>
<td>ALLOCATED</td>
</tr>
<tr>
<td>82</td>
<td>CONFFREE</td>
</tr>
<tr>
<td>83</td>
<td>CONFRECEIVE</td>
</tr>
<tr>
<td>84</td>
<td>CONFSEND</td>
</tr>
<tr>
<td>85</td>
<td>FREE</td>
</tr>
<tr>
<td>86</td>
<td>PENDFREE</td>
</tr>
<tr>
<td>87</td>
<td>PENDRECEIVE</td>
</tr>
<tr>
<td>88</td>
<td>RECEIVE</td>
</tr>
<tr>
<td>89</td>
<td>ROLLBACK</td>
</tr>
<tr>
<td>90</td>
<td>SEND</td>
</tr>
<tr>
<td>91</td>
<td>SYNCFREE</td>
</tr>
<tr>
<td>92</td>
<td>SYNCRECEIVE</td>
</tr>
<tr>
<td>93</td>
<td>SYNCSEND</td>
</tr>
<tr>
<td>94</td>
<td>FREEING</td>
</tr>
<tr>
<td>95</td>
<td>AVAILABLE</td>
</tr>
<tr>
<td>96</td>
<td>OBTAINING</td>
</tr>
<tr>
<td>100</td>
<td>NOTTABLE</td>
</tr>
<tr>
<td>101</td>
<td>CICSTABLE</td>
</tr>
<tr>
<td>102</td>
<td>USERTABLE</td>
</tr>
<tr>
<td>103</td>
<td>REMTABLE</td>
</tr>
<tr>
<td>110</td>
<td>PRIMARY</td>
</tr>
<tr>
<td>111</td>
<td>TAKEOVER</td>
</tr>
<tr>
<td>121</td>
<td>IRC</td>
</tr>
<tr>
<td>122</td>
<td>INDIRECT</td>
</tr>
<tr>
<td>123</td>
<td>XM</td>
</tr>
<tr>
<td>124</td>
<td>APPC</td>
</tr>
<tr>
<td>125</td>
<td>LU61</td>
</tr>
<tr>
<td>126</td>
<td>PENDING</td>
</tr>
<tr>
<td>127</td>
<td>NOTPENDING</td>
</tr>
<tr>
<td>143</td>
<td>XOK</td>
</tr>
<tr>
<td>144</td>
<td>XNOTDONE</td>
</tr>
<tr>
<td>147</td>
<td>JAVA</td>
</tr>
<tr>
<td>148</td>
<td>CPLUSPLUS</td>
</tr>
<tr>
<td>149</td>
<td>C</td>
</tr>
<tr>
<td>150</td>
<td>ASSEMBLER</td>
</tr>
<tr>
<td>151</td>
<td>COBOL</td>
</tr>
<tr>
<td>152</td>
<td>PLI</td>
</tr>
<tr>
<td>152</td>
<td>PL1</td>
</tr>
<tr>
<td>153</td>
<td>RPG</td>
</tr>
<tr>
<td>154</td>
<td>PROGRAM</td>
</tr>
<tr>
<td>155</td>
<td>MAP</td>
</tr>
<tr>
<td>155</td>
<td>MAPSET</td>
</tr>
<tr>
<td>156</td>
<td>PARTITIONSET</td>
</tr>
<tr>
<td>157</td>
<td>PROFILE</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>158</td>
<td>ANY</td>
</tr>
<tr>
<td>159</td>
<td>BELOW</td>
</tr>
<tr>
<td>160</td>
<td>PURGEABLE</td>
</tr>
<tr>
<td>161</td>
<td>NOTPURGEABLE</td>
</tr>
<tr>
<td>162</td>
<td>OLDCOPY</td>
</tr>
<tr>
<td>163</td>
<td>HOLD</td>
</tr>
<tr>
<td>164</td>
<td>NOHOLD</td>
</tr>
<tr>
<td>165</td>
<td>LPA</td>
</tr>
<tr>
<td>166</td>
<td>NOTLPA</td>
</tr>
<tr>
<td>167</td>
<td>NEWCOPY</td>
</tr>
<tr>
<td>168</td>
<td>PHASEIN</td>
</tr>
<tr>
<td>169</td>
<td>ALLCONN</td>
</tr>
<tr>
<td>170</td>
<td>AUTOCONN</td>
</tr>
<tr>
<td>171</td>
<td>NONAUTOCONN</td>
</tr>
<tr>
<td>172</td>
<td>GOINGOUT</td>
</tr>
<tr>
<td>173</td>
<td>SHARED</td>
</tr>
<tr>
<td>174</td>
<td>PRIVATE</td>
</tr>
<tr>
<td>175</td>
<td>SPRSTRACE</td>
</tr>
<tr>
<td>176</td>
<td>STANTRACE</td>
</tr>
<tr>
<td>177</td>
<td>SPECTRACE</td>
</tr>
<tr>
<td>178</td>
<td>DYNAMIC</td>
</tr>
<tr>
<td>179</td>
<td>STATIC</td>
</tr>
<tr>
<td>180</td>
<td>STARTUP</td>
</tr>
<tr>
<td>181</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>182</td>
<td>FIRSTQUIESCE</td>
</tr>
<tr>
<td>183</td>
<td>FINALQUIESCE</td>
</tr>
<tr>
<td>184</td>
<td>SYSDUMP</td>
</tr>
<tr>
<td>185</td>
<td>NOSYSDUMP</td>
</tr>
<tr>
<td>186</td>
<td>TRANDUMP</td>
</tr>
<tr>
<td>187</td>
<td>NOTRANDUMP</td>
</tr>
<tr>
<td>188</td>
<td>SWITCH</td>
</tr>
<tr>
<td>189</td>
<td>MODEANY</td>
</tr>
<tr>
<td>190</td>
<td>MODE24</td>
</tr>
<tr>
<td>191</td>
<td>MODE31</td>
</tr>
<tr>
<td>192</td>
<td>BACKOUT</td>
</tr>
<tr>
<td>193</td>
<td>NOBACKOUT</td>
</tr>
<tr>
<td>194</td>
<td>EXTSECURITY</td>
</tr>
<tr>
<td>195</td>
<td>CICSECURITY</td>
</tr>
<tr>
<td>196</td>
<td>NOSECURITY</td>
</tr>
<tr>
<td>197</td>
<td>ALTERNATE</td>
</tr>
<tr>
<td>198</td>
<td>DEFAULT</td>
</tr>
<tr>
<td>200</td>
<td>OFF</td>
</tr>
<tr>
<td>201</td>
<td>ON</td>
</tr>
<tr>
<td>202</td>
<td>RESSECNO</td>
</tr>
<tr>
<td>203</td>
<td>RESSECINT</td>
</tr>
<tr>
<td>204</td>
<td>RESSECEXT</td>
</tr>
<tr>
<td>204</td>
<td>RESSECYES</td>
</tr>
<tr>
<td>205</td>
<td>CMDSECNO</td>
</tr>
<tr>
<td>207</td>
<td>CMDSECEXT</td>
</tr>
<tr>
<td>207</td>
<td>CMDSECYES</td>
</tr>
<tr>
<td>208</td>
<td>COMMIT</td>
</tr>
<tr>
<td>209</td>
<td>READBACK</td>
</tr>
<tr>
<td>210</td>
<td>EMPTY</td>
</tr>
<tr>
<td>211</td>
<td>NOTEMPTY</td>
</tr>
</tbody>
</table>
Value  CVDA
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  FORCEREGULAR
238  FILE
239  TCLASS
240  TIME
241  MCHCTL
242  PRESETSEC
243  NOPRESETSEC
244  SIGNEDON
245  SIGNEDOFF
246  UOW
247  AUXILIARY
248  MAIN
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
<table>
<thead>
<tr>
<th>Value</th>
<th>CVDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>272</td>
<td>FOPEN</td>
</tr>
<tr>
<td>273</td>
<td>FCLOSE</td>
</tr>
<tr>
<td>274</td>
<td>OK</td>
</tr>
<tr>
<td>275</td>
<td>NOWRITE</td>
</tr>
<tr>
<td>275</td>
<td>READONLY</td>
</tr>
<tr>
<td>276</td>
<td>REMOVE</td>
</tr>
<tr>
<td>277</td>
<td>RECOVERED</td>
</tr>
<tr>
<td>282</td>
<td>AINIT</td>
</tr>
<tr>
<td>283</td>
<td>BINIT</td>
</tr>
<tr>
<td>284</td>
<td>AUTOINIT</td>
</tr>
<tr>
<td>285</td>
<td>NOSWITCH</td>
</tr>
<tr>
<td>286</td>
<td>SWITCHNEXT</td>
</tr>
<tr>
<td>287</td>
<td>SWITCHALL</td>
</tr>
<tr>
<td>288</td>
<td>SHUTDOWN</td>
</tr>
<tr>
<td>289</td>
<td>NOSHUTDOWN</td>
</tr>
<tr>
<td>290</td>
<td>RESET</td>
</tr>
<tr>
<td>291</td>
<td>ADD</td>
</tr>
<tr>
<td>292</td>
<td>DELETE</td>
</tr>
<tr>
<td>293</td>
<td>REUSE</td>
</tr>
<tr>
<td>294</td>
<td>NOREUSE</td>
</tr>
<tr>
<td>310</td>
<td>INTSTART</td>
</tr>
<tr>
<td>311</td>
<td>INTSTOP</td>
</tr>
<tr>
<td>312</td>
<td>AUXSTART</td>
</tr>
<tr>
<td>313</td>
<td>AUXPAUSE</td>
</tr>
<tr>
<td>314</td>
<td>AUXSTOP</td>
</tr>
<tr>
<td>317</td>
<td>GTFSTART</td>
</tr>
<tr>
<td>318</td>
<td>GTFSTOP</td>
</tr>
<tr>
<td>319</td>
<td>SYSTEMON</td>
</tr>
<tr>
<td>320</td>
<td>SYSTEMOFF</td>
</tr>
<tr>
<td>321</td>
<td>USERON</td>
</tr>
<tr>
<td>322</td>
<td>USEROFF</td>
</tr>
<tr>
<td>323</td>
<td>SINGLEON</td>
</tr>
<tr>
<td>324</td>
<td>SINGLEOFF</td>
</tr>
<tr>
<td>330</td>
<td>PERF</td>
</tr>
<tr>
<td>331</td>
<td>NOPERF</td>
</tr>
<tr>
<td>332</td>
<td>EXCEPT</td>
</tr>
<tr>
<td>333</td>
<td>NOEXCEPT</td>
</tr>
<tr>
<td>334</td>
<td>EVENT</td>
</tr>
<tr>
<td>335</td>
<td>NOEVENT</td>
</tr>
<tr>
<td>336</td>
<td>RESRCE</td>
</tr>
<tr>
<td>337</td>
<td>NORESRCE</td>
</tr>
<tr>
<td>338</td>
<td>APPLNAME</td>
</tr>
<tr>
<td>339</td>
<td>NOAPPLNAME</td>
</tr>
<tr>
<td>340</td>
<td>WAIT</td>
</tr>
<tr>
<td>341</td>
<td>NOWAIT</td>
</tr>
<tr>
<td>342</td>
<td>FORCE</td>
</tr>
<tr>
<td>342</td>
<td>FORCEUOW</td>
</tr>
<tr>
<td>349</td>
<td>CLOSEFAILED</td>
</tr>
<tr>
<td>350</td>
<td>IMMCLOSE</td>
</tr>
<tr>
<td>351</td>
<td>FORCECLOSE</td>
</tr>
<tr>
<td>352</td>
<td>IMMCLOSING</td>
</tr>
<tr>
<td>353</td>
<td>FORCENCLOSING</td>
</tr>
<tr>
<td>354</td>
<td>FWDRECOVABLE</td>
</tr>
<tr>
<td>355</td>
<td>UNDETERMINED</td>
</tr>
<tr>
<td>Value</td>
<td>CVDA</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------</td>
</tr>
<tr>
<td>356</td>
<td>NORMALBKOUT</td>
</tr>
<tr>
<td>357</td>
<td>FAILEDBKOUT</td>
</tr>
<tr>
<td>358</td>
<td>FAILINGBKOUT</td>
</tr>
<tr>
<td>359</td>
<td>INVALID</td>
</tr>
<tr>
<td>360</td>
<td>VALID</td>
</tr>
<tr>
<td>361</td>
<td>NOTFWDRCVBLE</td>
</tr>
<tr>
<td>362</td>
<td>EXITTRACE</td>
</tr>
<tr>
<td>363</td>
<td>NOEXITTRACE</td>
</tr>
<tr>
<td>364</td>
<td>ZCPTRACE</td>
</tr>
<tr>
<td>365</td>
<td>NOZCPTRACE</td>
</tr>
<tr>
<td>366</td>
<td>TCEXITALL</td>
</tr>
<tr>
<td>367</td>
<td>TCEXITSYSTEM</td>
</tr>
<tr>
<td>368</td>
<td>TCEXITNONE</td>
</tr>
<tr>
<td>369</td>
<td>TCEXITALLOFF</td>
</tr>
<tr>
<td>370</td>
<td>CEDF</td>
</tr>
<tr>
<td>371</td>
<td>MODEL</td>
</tr>
<tr>
<td>372</td>
<td>NOCEDF</td>
</tr>
<tr>
<td>373</td>
<td>SURROGATE</td>
</tr>
<tr>
<td>374</td>
<td>SESSION</td>
</tr>
<tr>
<td>375</td>
<td>APPCSINGLE</td>
</tr>
<tr>
<td>376</td>
<td>APPCPARALLEL</td>
</tr>
<tr>
<td>377</td>
<td>COBOLII</td>
</tr>
<tr>
<td>378</td>
<td>NOTINIT</td>
</tr>
<tr>
<td>379</td>
<td>LE370</td>
</tr>
<tr>
<td>380</td>
<td>INACTIVE</td>
</tr>
<tr>
<td>381</td>
<td>CICSDATAKEY</td>
</tr>
<tr>
<td>382</td>
<td>USERDATAKEY</td>
</tr>
<tr>
<td>383</td>
<td>CICSEXECKEY</td>
</tr>
<tr>
<td>384</td>
<td>USEREXECKEY</td>
</tr>
<tr>
<td>385</td>
<td>DPLSUBSET</td>
</tr>
<tr>
<td>386</td>
<td>FULLAPI</td>
</tr>
<tr>
<td>387</td>
<td>FMHPARM</td>
</tr>
<tr>
<td>388</td>
<td>NOFMHPARM</td>
</tr>
<tr>
<td>389</td>
<td>OBOPERID</td>
</tr>
<tr>
<td>390</td>
<td>NOOBOPERID</td>
</tr>
<tr>
<td>391</td>
<td>APLKYBD</td>
</tr>
<tr>
<td>392</td>
<td>NOAPLKYBD</td>
</tr>
<tr>
<td>393</td>
<td>APLTEXT</td>
</tr>
<tr>
<td>394</td>
<td>NOAPLTEXT</td>
</tr>
<tr>
<td>395</td>
<td>AUDALARM</td>
</tr>
<tr>
<td>396</td>
<td>NOAUDALARM</td>
</tr>
<tr>
<td>397</td>
<td>BACKTRANS</td>
</tr>
<tr>
<td>398</td>
<td>NOBACKTRANS</td>
</tr>
<tr>
<td>399</td>
<td>COLOR</td>
</tr>
<tr>
<td>400</td>
<td>NOCOLOR</td>
</tr>
<tr>
<td>401</td>
<td>COPY</td>
</tr>
<tr>
<td>402</td>
<td>NOCOPY</td>
</tr>
<tr>
<td>403</td>
<td>DUALCASE</td>
</tr>
<tr>
<td>404</td>
<td>NODUALCASE</td>
</tr>
<tr>
<td>405</td>
<td>EXTENDEDDDS</td>
</tr>
<tr>
<td>406</td>
<td>NOEXTENDEDDDS</td>
</tr>
<tr>
<td>407</td>
<td>FORMFEED</td>
</tr>
<tr>
<td>408</td>
<td>NOFORMFEED</td>
</tr>
<tr>
<td>409</td>
<td>HFORM</td>
</tr>
<tr>
<td>Value</td>
<td>CVDA</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>410</td>
<td>NOFORM</td>
</tr>
<tr>
<td>411</td>
<td>VFORM</td>
</tr>
<tr>
<td>412</td>
<td>NOVFORM</td>
</tr>
<tr>
<td>413</td>
<td>HILIGHT</td>
</tr>
<tr>
<td>414</td>
<td>NOHILIGHT</td>
</tr>
<tr>
<td>415</td>
<td>KATAKANA</td>
</tr>
<tr>
<td>416</td>
<td>NOKATAKANA</td>
</tr>
<tr>
<td>417</td>
<td>LIGHTPEN</td>
</tr>
<tr>
<td>418</td>
<td>NOLIGHTPEN</td>
</tr>
<tr>
<td>419</td>
<td>MSRCONTROL</td>
</tr>
<tr>
<td>420</td>
<td>NOMSRCONTROL</td>
</tr>
<tr>
<td>421</td>
<td>OBFORMAT</td>
</tr>
<tr>
<td>422</td>
<td>NOOBFORMAT</td>
</tr>
<tr>
<td>423</td>
<td>OUTLINE</td>
</tr>
<tr>
<td>424</td>
<td>NOOUTLINE</td>
</tr>
<tr>
<td>425</td>
<td>PARTITIONS</td>
</tr>
<tr>
<td>426</td>
<td>NOPARTITIONS</td>
</tr>
<tr>
<td>427</td>
<td>PRINTADAPT</td>
</tr>
<tr>
<td>428</td>
<td>NOPRINTADAPT</td>
</tr>
<tr>
<td>429</td>
<td>PROGSYMBOL</td>
</tr>
<tr>
<td>430</td>
<td>NOPROGSYMBOL</td>
</tr>
<tr>
<td>431</td>
<td>ALLQUERY</td>
</tr>
<tr>
<td>432</td>
<td>NOQUERY</td>
</tr>
<tr>
<td>433</td>
<td>COLDQUERY</td>
</tr>
<tr>
<td>434</td>
<td>SOSI</td>
</tr>
<tr>
<td>435</td>
<td>NOSOSI</td>
</tr>
<tr>
<td>436</td>
<td>TEXTKYBD</td>
</tr>
<tr>
<td>437</td>
<td>NOTEXTKYBD</td>
</tr>
<tr>
<td>438</td>
<td>TEXTPRINT</td>
</tr>
<tr>
<td>439</td>
<td>NOTEXTPRINT</td>
</tr>
<tr>
<td>440</td>
<td>VALIDATION</td>
</tr>
<tr>
<td>441</td>
<td>NOVALIDATION</td>
</tr>
<tr>
<td>442</td>
<td>RELREQ</td>
</tr>
<tr>
<td>443</td>
<td>NORELREQ</td>
</tr>
<tr>
<td>444</td>
<td>DISCREQ</td>
</tr>
<tr>
<td>445</td>
<td>NODISCREQ</td>
</tr>
<tr>
<td>446</td>
<td>ALTPRTCOPY</td>
</tr>
<tr>
<td>447</td>
<td>NOALTPRTCOPY</td>
</tr>
<tr>
<td>448</td>
<td>PRTCOPY</td>
</tr>
<tr>
<td>449</td>
<td>NOPRTCOPY</td>
</tr>
<tr>
<td>450</td>
<td>UCTRAN</td>
</tr>
<tr>
<td>451</td>
<td>NOUCTRAN</td>
</tr>
<tr>
<td>452</td>
<td>TRANIDONLY</td>
</tr>
<tr>
<td>483</td>
<td>TEMPORARY</td>
</tr>
<tr>
<td>484</td>
<td>PERMANENT</td>
</tr>
<tr>
<td>485</td>
<td>NEWSESSION</td>
</tr>
<tr>
<td>486</td>
<td>OLDESESSION</td>
</tr>
<tr>
<td>487</td>
<td>NOSTSN</td>
</tr>
<tr>
<td>488</td>
<td>STSNSET</td>
</tr>
<tr>
<td>489</td>
<td>STSNTST</td>
</tr>
<tr>
<td>490</td>
<td>EB</td>
</tr>
<tr>
<td>491</td>
<td>CD</td>
</tr>
<tr>
<td>492</td>
<td>MORE</td>
</tr>
<tr>
<td>493</td>
<td>LIC</td>
</tr>
<tr>
<td>Value</td>
<td>CVDA</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>494</td>
<td>RU</td>
</tr>
<tr>
<td>495</td>
<td>NOTCDEB</td>
</tr>
<tr>
<td>496</td>
<td>NONE</td>
</tr>
<tr>
<td>497</td>
<td>DEFRESP1</td>
</tr>
<tr>
<td>498</td>
<td>DEFRESP2</td>
</tr>
<tr>
<td>499</td>
<td>DEFRESP3</td>
</tr>
<tr>
<td>500</td>
<td>NOALARM</td>
</tr>
<tr>
<td>501</td>
<td>ALARM</td>
</tr>
<tr>
<td>502</td>
<td>FMH</td>
</tr>
<tr>
<td>503</td>
<td>NOFMH</td>
</tr>
<tr>
<td>504</td>
<td>PROTECTED</td>
</tr>
<tr>
<td>505</td>
<td>UNPROTECTED</td>
</tr>
<tr>
<td>506</td>
<td>MDT</td>
</tr>
<tr>
<td>507</td>
<td>NOMDT</td>
</tr>
<tr>
<td>508</td>
<td>DATA</td>
</tr>
<tr>
<td>509</td>
<td>STSN</td>
</tr>
<tr>
<td>510</td>
<td>BEGINSESSION</td>
</tr>
<tr>
<td>511</td>
<td>TIMEOUT</td>
</tr>
<tr>
<td>512</td>
<td>INSTALLFAIL</td>
</tr>
<tr>
<td>513</td>
<td>DISCARDFAIL</td>
</tr>
<tr>
<td>514</td>
<td>SETFAIL</td>
</tr>
<tr>
<td>515</td>
<td>ACOFAIL</td>
</tr>
<tr>
<td>516</td>
<td>SESSIONLOST</td>
</tr>
<tr>
<td>517</td>
<td>SESSIONFAIL</td>
</tr>
<tr>
<td>518</td>
<td>CONVIDLE</td>
</tr>
<tr>
<td>519</td>
<td>ADDFAIL</td>
</tr>
<tr>
<td>520</td>
<td>DELETEFAIL</td>
</tr>
<tr>
<td>521</td>
<td>UNSOLDATA</td>
</tr>
<tr>
<td>522</td>
<td>NORMALRESP</td>
</tr>
<tr>
<td>523</td>
<td>EXCEPTRESP</td>
</tr>
<tr>
<td>524</td>
<td>ATTENTION</td>
</tr>
<tr>
<td>525</td>
<td>LUSTAT</td>
</tr>
<tr>
<td>526</td>
<td>CANCEL</td>
</tr>
<tr>
<td>527</td>
<td>RTR</td>
</tr>
<tr>
<td>528</td>
<td>DEFRESP1OR2</td>
</tr>
<tr>
<td>529</td>
<td>POSITIVE</td>
</tr>
<tr>
<td>530</td>
<td>NEGATIVE</td>
</tr>
<tr>
<td>531</td>
<td>NOMSGJRNL</td>
</tr>
<tr>
<td>532</td>
<td>INOUT</td>
</tr>
<tr>
<td>533</td>
<td>T3278M2</td>
</tr>
<tr>
<td>534</td>
<td>T3278M3</td>
</tr>
<tr>
<td>535</td>
<td>T3278M4</td>
</tr>
<tr>
<td>536</td>
<td>T3278M5</td>
</tr>
<tr>
<td>537</td>
<td>T3279M2</td>
</tr>
<tr>
<td>538</td>
<td>T3279M3</td>
</tr>
<tr>
<td>539</td>
<td>T3279M4</td>
</tr>
<tr>
<td>540</td>
<td>T3279M5</td>
</tr>
<tr>
<td>541</td>
<td>LUP</td>
</tr>
<tr>
<td>542</td>
<td>FORMATTED</td>
</tr>
<tr>
<td>543</td>
<td>DATASTREAM</td>
</tr>
<tr>
<td>544</td>
<td>LOSE</td>
</tr>
<tr>
<td>545</td>
<td>WIN</td>
</tr>
<tr>
<td>546</td>
<td>NOTINBOUND</td>
</tr>
<tr>
<td>547</td>
<td>INBOUND</td>
</tr>
<tr>
<td>Value</td>
<td>CVDA</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>549</td>
<td>RELEASING</td>
</tr>
<tr>
<td>550</td>
<td>INSTALLED</td>
</tr>
<tr>
<td>551</td>
<td>NOTINSTALLED</td>
</tr>
<tr>
<td>552</td>
<td>TPS55M2</td>
</tr>
<tr>
<td>553</td>
<td>TPS55M3</td>
</tr>
<tr>
<td>554</td>
<td>TPS55M4</td>
</tr>
<tr>
<td>555</td>
<td>TPS55M5</td>
</tr>
<tr>
<td>556</td>
<td>NOCONV</td>
</tr>
<tr>
<td>557</td>
<td>PENDSTSN</td>
</tr>
<tr>
<td>558</td>
<td>PENDBEGIN</td>
</tr>
<tr>
<td>559</td>
<td>APPLICATION</td>
</tr>
<tr>
<td>560</td>
<td>PENDDATA</td>
</tr>
<tr>
<td>561</td>
<td>PENDSTART</td>
</tr>
<tr>
<td>562</td>
<td>PENDRELEASE</td>
</tr>
<tr>
<td>563</td>
<td>RELEASE</td>
</tr>
<tr>
<td>564</td>
<td>PENDUNSOL</td>
</tr>
<tr>
<td>565</td>
<td>PENDPASS</td>
</tr>
<tr>
<td>600</td>
<td>CONVERSE</td>
</tr>
<tr>
<td>601</td>
<td>NOCONVERSE</td>
</tr>
<tr>
<td>602</td>
<td>SYNCPOINT</td>
</tr>
<tr>
<td>603</td>
<td>NOSYNCPPOINT</td>
</tr>
<tr>
<td>604</td>
<td>GMT</td>
</tr>
<tr>
<td>605</td>
<td>LOCAL</td>
</tr>
<tr>
<td>606</td>
<td>FORMATEDF</td>
</tr>
<tr>
<td>607</td>
<td>NOFORMATEDF</td>
</tr>
<tr>
<td>608</td>
<td>NOTASKSTART</td>
</tr>
<tr>
<td>609</td>
<td>STARTED</td>
</tr>
<tr>
<td>610</td>
<td>STOPPED</td>
</tr>
<tr>
<td>611</td>
<td>TASKSTART</td>
</tr>
<tr>
<td>612</td>
<td>BUSY</td>
</tr>
<tr>
<td>613</td>
<td>NOTBUSY</td>
</tr>
<tr>
<td>614</td>
<td>SCS</td>
</tr>
<tr>
<td>615</td>
<td>DS3270</td>
</tr>
<tr>
<td>616</td>
<td>ASCII7</td>
</tr>
<tr>
<td>617</td>
<td>ASCII8</td>
</tr>
<tr>
<td>618</td>
<td>AUTOSTART</td>
</tr>
<tr>
<td>620</td>
<td>INDOUBT</td>
</tr>
<tr>
<td>621</td>
<td>INFLIGHT</td>
</tr>
<tr>
<td>622</td>
<td>WAITFORGET</td>
</tr>
<tr>
<td>623</td>
<td>CONTROLSHUT</td>
</tr>
<tr>
<td>624</td>
<td>CANCELLED</td>
</tr>
<tr>
<td>625</td>
<td>FIRSTINIT</td>
</tr>
<tr>
<td>626</td>
<td>SECONDOINIT</td>
</tr>
<tr>
<td>627</td>
<td>THIRDINIT</td>
</tr>
<tr>
<td>628</td>
<td>INITCOMPLETE</td>
</tr>
<tr>
<td>629</td>
<td>STANDBY</td>
</tr>
<tr>
<td>630</td>
<td>AUTOACTIVE</td>
</tr>
<tr>
<td>631</td>
<td>AUTOINACTIVE</td>
</tr>
<tr>
<td>632</td>
<td>CTLGALL</td>
</tr>
<tr>
<td>633</td>
<td>CTLGMODIFY</td>
</tr>
<tr>
<td>634</td>
<td>CTLGNONE</td>
</tr>
<tr>
<td>635</td>
<td>START</td>
</tr>
<tr>
<td>636</td>
<td>POST</td>
</tr>
<tr>
<td>637</td>
<td>DELAY</td>
</tr>
</tbody>
</table>

Appendix A. CICS-value data areas used by all commands
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>638</td>
<td>ROUTE</td>
</tr>
<tr>
<td>640</td>
<td>CLEAR</td>
</tr>
<tr>
<td>641</td>
<td>NOCLEAR</td>
</tr>
<tr>
<td>642</td>
<td>USER</td>
</tr>
<tr>
<td>643</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>644</td>
<td>SHUTENABLED</td>
</tr>
<tr>
<td>645</td>
<td>SHUTDISABLED</td>
</tr>
<tr>
<td>646</td>
<td>RFC1123</td>
</tr>
<tr>
<td>650</td>
<td>EXCI</td>
</tr>
<tr>
<td>651</td>
<td>GENERIC</td>
</tr>
<tr>
<td>652</td>
<td>SPECIFIC</td>
</tr>
<tr>
<td>653</td>
<td>SYSCONNECT</td>
</tr>
<tr>
<td>654</td>
<td>NOSYSCONNECT</td>
</tr>
<tr>
<td>655</td>
<td>FORCEDCANCEL</td>
</tr>
<tr>
<td>657</td>
<td>NOISOLATE</td>
</tr>
<tr>
<td>658</td>
<td>ISOLATE</td>
</tr>
<tr>
<td>659</td>
<td>NOTDEFINED</td>
</tr>
<tr>
<td>660</td>
<td>CICS</td>
</tr>
<tr>
<td>661</td>
<td>NONCICS</td>
</tr>
<tr>
<td>662</td>
<td>SYSPLEX</td>
</tr>
<tr>
<td>663</td>
<td>SUBSPACE</td>
</tr>
<tr>
<td>664</td>
<td>BASESPACE</td>
</tr>
<tr>
<td>665</td>
<td>XCF</td>
</tr>
<tr>
<td>666</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>667</td>
<td>NOTREQUIRED</td>
</tr>
<tr>
<td>668</td>
<td>SOS</td>
</tr>
<tr>
<td>669</td>
<td>NOTSOS</td>
</tr>
<tr>
<td>670</td>
<td>REGISTERED</td>
</tr>
<tr>
<td>671</td>
<td>UNREGISTERED</td>
</tr>
<tr>
<td>672</td>
<td>UNAVAILABLE</td>
</tr>
<tr>
<td>673</td>
<td>CMDPROT</td>
</tr>
<tr>
<td>674</td>
<td>NOCMDPROT</td>
</tr>
<tr>
<td>675</td>
<td>RELATED</td>
</tr>
<tr>
<td>677</td>
<td>REGERROR</td>
</tr>
<tr>
<td>678</td>
<td>DEREGISTERED</td>
</tr>
<tr>
<td>679</td>
<td>DEREGERROR</td>
</tr>
<tr>
<td>680</td>
<td>REENTPROT</td>
</tr>
<tr>
<td>681</td>
<td>NOREENTPROT</td>
</tr>
<tr>
<td>682</td>
<td>SOSBELOW</td>
</tr>
<tr>
<td>683</td>
<td>SOSABOVE</td>
</tr>
<tr>
<td>684</td>
<td>DAE</td>
</tr>
<tr>
<td>685</td>
<td>NODAE</td>
</tr>
<tr>
<td>690</td>
<td>CONNECTED</td>
</tr>
<tr>
<td>691</td>
<td>NOTCONNECTED</td>
</tr>
<tr>
<td>693</td>
<td>SPI</td>
</tr>
<tr>
<td>694</td>
<td>NOSPI</td>
</tr>
<tr>
<td>700</td>
<td>NORECOVDATA</td>
</tr>
<tr>
<td>701</td>
<td>RECOVDATA</td>
</tr>
<tr>
<td>702</td>
<td>RESYNC</td>
</tr>
<tr>
<td>703</td>
<td>UNCONNECTED</td>
</tr>
<tr>
<td>706</td>
<td>IMMQUIESCED</td>
</tr>
<tr>
<td>707</td>
<td>QUIESCED</td>
</tr>
<tr>
<td>708</td>
<td>QUIESCING</td>
</tr>
<tr>
<td>709</td>
<td>UNQUIESCED</td>
</tr>
</tbody>
</table>
### Appendix A. CICS-value data areas used by all commands

<table>
<thead>
<tr>
<th>Value</th>
<th>CVDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>710</td>
<td>NOLOSTLOCKS</td>
</tr>
<tr>
<td>711</td>
<td>NORETAINED</td>
</tr>
<tr>
<td>712</td>
<td>RECOVERLOCKS</td>
</tr>
<tr>
<td>713</td>
<td>REMLOSTLOCKS</td>
</tr>
<tr>
<td>714</td>
<td>RESETLOCKS</td>
</tr>
<tr>
<td>715</td>
<td>RETAINED</td>
</tr>
<tr>
<td>716</td>
<td>RETRY</td>
</tr>
<tr>
<td>720</td>
<td>RLS</td>
</tr>
<tr>
<td>721</td>
<td>NOTRLS</td>
</tr>
<tr>
<td>722</td>
<td>UNCOMMITTED</td>
</tr>
<tr>
<td>723</td>
<td>CONSISTENT</td>
</tr>
<tr>
<td>724</td>
<td>REPEATABLE</td>
</tr>
<tr>
<td>730</td>
<td>RLSACTIVE</td>
</tr>
<tr>
<td>731</td>
<td>RLSINACTIVE</td>
</tr>
<tr>
<td>732</td>
<td>VRRDS</td>
</tr>
<tr>
<td>735</td>
<td>CHUNKNO</td>
</tr>
<tr>
<td>736</td>
<td>CHUNKYES</td>
</tr>
<tr>
<td>737</td>
<td>CHUNKEND</td>
</tr>
<tr>
<td>738</td>
<td>SRVCONVERT</td>
</tr>
<tr>
<td>739</td>
<td>NOSRVCONVERT</td>
</tr>
<tr>
<td>740</td>
<td>REMSESSION</td>
</tr>
<tr>
<td>741</td>
<td>CLOSE</td>
</tr>
<tr>
<td>742</td>
<td>NOCLOSE</td>
</tr>
<tr>
<td>743</td>
<td>CLICONVERT</td>
</tr>
<tr>
<td>744</td>
<td>NOCLICONVERT</td>
</tr>
<tr>
<td>745</td>
<td>NOINCONVERT</td>
</tr>
<tr>
<td>746</td>
<td>NOOUTCONVERT</td>
</tr>
<tr>
<td>747</td>
<td>EXPECT</td>
</tr>
<tr>
<td>748</td>
<td>GET</td>
</tr>
<tr>
<td>749</td>
<td>PUT</td>
</tr>
<tr>
<td>750</td>
<td>HEAD</td>
</tr>
<tr>
<td>751</td>
<td>EXECENQ</td>
</tr>
<tr>
<td>752</td>
<td>EXECENQADDR</td>
</tr>
<tr>
<td>753</td>
<td>OWNER</td>
</tr>
<tr>
<td>754</td>
<td>WAITER</td>
</tr>
<tr>
<td>755</td>
<td>CONNECTION</td>
</tr>
<tr>
<td>756</td>
<td>DATASET</td>
</tr>
<tr>
<td>757</td>
<td>HEURCOMMIT</td>
</tr>
<tr>
<td>758</td>
<td>HEURBACKOUT</td>
</tr>
<tr>
<td>759</td>
<td>OPTIONS</td>
</tr>
<tr>
<td>760</td>
<td>TRACE</td>
</tr>
<tr>
<td>761</td>
<td>RLSSERVER</td>
</tr>
<tr>
<td>762</td>
<td>SHUNTED</td>
</tr>
<tr>
<td>763</td>
<td>WAITCOMMIT</td>
</tr>
<tr>
<td>765</td>
<td>WAITING</td>
</tr>
<tr>
<td>766</td>
<td>WAITRMI</td>
</tr>
<tr>
<td>767</td>
<td>TDQ</td>
</tr>
<tr>
<td>768</td>
<td>TSQ</td>
</tr>
<tr>
<td>770</td>
<td>COORDINATOR</td>
</tr>
<tr>
<td>771</td>
<td>RMI</td>
</tr>
<tr>
<td>772</td>
<td>STARTING</td>
</tr>
<tr>
<td>773</td>
<td>SUBORDINATE</td>
</tr>
<tr>
<td>774</td>
<td>NRS</td>
</tr>
<tr>
<td>774</td>
<td>UNKNOWN</td>
</tr>
<tr>
<td>Value</td>
<td>CVDA</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>775</td>
<td>NORMI</td>
</tr>
<tr>
<td>780</td>
<td>MVS</td>
</tr>
<tr>
<td>781</td>
<td>DUMMY</td>
</tr>
<tr>
<td>782</td>
<td>FAILED</td>
</tr>
<tr>
<td>783</td>
<td>FLUSH</td>
</tr>
<tr>
<td>784</td>
<td>NOSYSLOG</td>
</tr>
<tr>
<td>785</td>
<td>SYSLOG</td>
</tr>
<tr>
<td>786</td>
<td>KEEP</td>
</tr>
<tr>
<td>788</td>
<td>COLD</td>
</tr>
<tr>
<td>789</td>
<td>INITIAL</td>
</tr>
<tr>
<td>790</td>
<td>ENDAFFINITY</td>
</tr>
<tr>
<td>791</td>
<td>CACHE</td>
</tr>
<tr>
<td>792</td>
<td>COMMITFAIL</td>
</tr>
<tr>
<td>793</td>
<td>DATASETFULL</td>
</tr>
<tr>
<td>794</td>
<td>DEADLOCK</td>
</tr>
<tr>
<td>795</td>
<td>DELEXITERROR</td>
</tr>
<tr>
<td>796</td>
<td>INDEXRECFULL</td>
</tr>
<tr>
<td>797</td>
<td>IOERROR</td>
</tr>
<tr>
<td>798</td>
<td>OPENERROR</td>
</tr>
<tr>
<td>799</td>
<td>RLGSNONE</td>
</tr>
<tr>
<td>800</td>
<td>BACKUPNONBWO</td>
</tr>
<tr>
<td>810</td>
<td>SKIP</td>
</tr>
<tr>
<td>811</td>
<td>LEAVE</td>
</tr>
<tr>
<td>811</td>
<td>REWIND</td>
</tr>
<tr>
<td>812</td>
<td>REREAD</td>
</tr>
<tr>
<td>813</td>
<td>MOD</td>
</tr>
<tr>
<td>814</td>
<td>QUEUE</td>
</tr>
<tr>
<td>815</td>
<td>REJECT</td>
</tr>
<tr>
<td>816</td>
<td>IGNORERR</td>
</tr>
<tr>
<td>820</td>
<td>UNATTEMPTED</td>
</tr>
<tr>
<td>830</td>
<td>RRcommitFAIL</td>
</tr>
<tr>
<td>831</td>
<td>RRINDOUBT</td>
</tr>
<tr>
<td>832</td>
<td>LCKSTRUCFAIL</td>
</tr>
<tr>
<td>833</td>
<td>CFTABLE</td>
</tr>
<tr>
<td>834</td>
<td>LOAD</td>
</tr>
<tr>
<td>835</td>
<td>NOLOAD</td>
</tr>
<tr>
<td>836</td>
<td>CONTENTION</td>
</tr>
<tr>
<td>837</td>
<td>LOCKING</td>
</tr>
<tr>
<td>900</td>
<td>ABEND</td>
</tr>
<tr>
<td>902</td>
<td>CGROUP</td>
</tr>
<tr>
<td>903</td>
<td>CONNECT</td>
</tr>
<tr>
<td>904</td>
<td>CONNECTING</td>
</tr>
<tr>
<td>905</td>
<td>CSIGN</td>
</tr>
<tr>
<td>906</td>
<td>CTERM</td>
</tr>
<tr>
<td>907</td>
<td>CTX</td>
</tr>
<tr>
<td>908</td>
<td>COPID</td>
</tr>
<tr>
<td>909</td>
<td>CUSERID</td>
</tr>
<tr>
<td>910</td>
<td>DISCONNING</td>
</tr>
<tr>
<td>911</td>
<td>EQUAL</td>
</tr>
<tr>
<td>912</td>
<td>EXIT</td>
</tr>
<tr>
<td>913</td>
<td>GROUP</td>
</tr>
<tr>
<td>914</td>
<td>HIGH</td>
</tr>
<tr>
<td>915</td>
<td>LOW</td>
</tr>
<tr>
<td>916</td>
<td>NO connect</td>
</tr>
<tr>
<td>Value</td>
<td>CVDA</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------</td>
</tr>
<tr>
<td>917</td>
<td>NOEXIT</td>
</tr>
<tr>
<td>918</td>
<td>NORELEASE</td>
</tr>
<tr>
<td>919</td>
<td>NOROLLBACK</td>
</tr>
<tr>
<td>920</td>
<td>NOTWAIT</td>
</tr>
<tr>
<td>921</td>
<td>N906D</td>
</tr>
<tr>
<td>922</td>
<td>POOL</td>
</tr>
<tr>
<td>923</td>
<td>REBUILD</td>
</tr>
<tr>
<td>924</td>
<td>RECONNECT</td>
</tr>
<tr>
<td>925</td>
<td>SIGN</td>
</tr>
<tr>
<td>926</td>
<td>SQLCODE</td>
</tr>
<tr>
<td>927</td>
<td>TWAIT</td>
</tr>
<tr>
<td>928</td>
<td>TXID</td>
</tr>
<tr>
<td>929</td>
<td>TX</td>
</tr>
<tr>
<td>930</td>
<td>USERID</td>
</tr>
<tr>
<td>931</td>
<td>N906</td>
</tr>
<tr>
<td>932</td>
<td>TPOOL</td>
</tr>
<tr>
<td>933</td>
<td>OPID</td>
</tr>
<tr>
<td>934</td>
<td>NORESYNC</td>
</tr>
<tr>
<td>1000</td>
<td>NOTFIRED</td>
</tr>
<tr>
<td>1001</td>
<td>FIRED</td>
</tr>
<tr>
<td>1002</td>
<td>ACTIVITY</td>
</tr>
<tr>
<td>1003</td>
<td>COMPOSITE</td>
</tr>
<tr>
<td>1004</td>
<td>TIMER</td>
</tr>
<tr>
<td>1005</td>
<td>AND</td>
</tr>
<tr>
<td>1006</td>
<td>OR</td>
</tr>
<tr>
<td>1010</td>
<td>PROCESS</td>
</tr>
<tr>
<td>1011</td>
<td>NOUSER</td>
</tr>
<tr>
<td>1012</td>
<td>EVENTFAIL</td>
</tr>
<tr>
<td>1013</td>
<td>FORCED</td>
</tr>
<tr>
<td>1014</td>
<td>INCOMPLETE</td>
</tr>
<tr>
<td>1015</td>
<td>UNEXPECTED</td>
</tr>
<tr>
<td>1016</td>
<td>NORMAL</td>
</tr>
<tr>
<td>1017</td>
<td>EXPIRED</td>
</tr>
<tr>
<td>1018</td>
<td>UNEXPIRED</td>
</tr>
<tr>
<td>1019</td>
<td>CHAR</td>
</tr>
<tr>
<td>1020</td>
<td>BIT</td>
</tr>
<tr>
<td>1021</td>
<td>NOTDYNAMIC</td>
</tr>
<tr>
<td>1022</td>
<td>NOTROUTABLE</td>
</tr>
<tr>
<td>1023</td>
<td>ROUTABLE</td>
</tr>
<tr>
<td>1024</td>
<td>DORMANT</td>
</tr>
<tr>
<td>1025</td>
<td>CANCELLING</td>
</tr>
<tr>
<td>1026</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>1027</td>
<td>NOTSUSPENDED</td>
</tr>
<tr>
<td>1029</td>
<td>HTTPS</td>
</tr>
<tr>
<td>1030</td>
<td>SSL</td>
</tr>
<tr>
<td>1031</td>
<td>NOSSL</td>
</tr>
<tr>
<td>1032</td>
<td>CLIENTAUTH</td>
</tr>
<tr>
<td>1033</td>
<td>HTTPYES</td>
</tr>
<tr>
<td>1034</td>
<td>HTTPNO</td>
</tr>
<tr>
<td>1035</td>
<td>PDSMEMBER</td>
</tr>
<tr>
<td>1036</td>
<td>APPEND</td>
</tr>
<tr>
<td>1037</td>
<td>NOAPPEND</td>
</tr>
<tr>
<td>1038</td>
<td>BINARY</td>
</tr>
<tr>
<td>1039</td>
<td>EBCDIC</td>
</tr>
<tr>
<td>Value</td>
<td>CVDA</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>1040</td>
<td>HFSFILE</td>
</tr>
<tr>
<td>1044</td>
<td>SOAP</td>
</tr>
<tr>
<td>1045</td>
<td>SENDER</td>
</tr>
<tr>
<td>1046</td>
<td>RECEIVER</td>
</tr>
<tr>
<td>1047</td>
<td>CHANNEL</td>
</tr>
<tr>
<td>1048</td>
<td>COMMAREA</td>
</tr>
<tr>
<td>1049</td>
<td>UPDATING</td>
</tr>
<tr>
<td>1050</td>
<td>QUASIRENT</td>
</tr>
<tr>
<td>1051</td>
<td>THREDDSAFE</td>
</tr>
<tr>
<td>1052</td>
<td>BASEAPI</td>
</tr>
<tr>
<td>1053</td>
<td>CICSAPI</td>
</tr>
<tr>
<td>1054</td>
<td>OPENAPI</td>
</tr>
<tr>
<td>1055</td>
<td>CKOPEN</td>
</tr>
<tr>
<td>1056</td>
<td>UKOPEN</td>
</tr>
<tr>
<td>1057</td>
<td>QR</td>
</tr>
<tr>
<td>1058</td>
<td>INTERNAL</td>
</tr>
<tr>
<td>1063</td>
<td>RRMS</td>
</tr>
<tr>
<td>1068</td>
<td>XPLINK</td>
</tr>
<tr>
<td>1071</td>
<td>NOAUTO</td>
</tr>
<tr>
<td>1072</td>
<td>PROGAUTO</td>
</tr>
<tr>
<td>1073</td>
<td>FULLAUTO</td>
</tr>
<tr>
<td>1075</td>
<td>CLASSCACHE</td>
</tr>
<tr>
<td>1076</td>
<td>NOCLASSCACHE</td>
</tr>
<tr>
<td>1077</td>
<td>TERMINATE</td>
</tr>
<tr>
<td>1078</td>
<td>RELOAD</td>
</tr>
<tr>
<td>1079</td>
<td>RELOADING</td>
</tr>
<tr>
<td>1080</td>
<td>JVM</td>
</tr>
<tr>
<td>1081</td>
<td>NOJVM</td>
</tr>
<tr>
<td>1082</td>
<td>DEBUG</td>
</tr>
<tr>
<td>1083</td>
<td>NODEBUG</td>
</tr>
<tr>
<td>1084</td>
<td>NONLE370</td>
</tr>
<tr>
<td>1085</td>
<td>PUBLISH</td>
</tr>
<tr>
<td>1086</td>
<td>RETRACT</td>
</tr>
<tr>
<td>1087</td>
<td>CLIENTCERT</td>
</tr>
<tr>
<td>1088</td>
<td>SCAN</td>
</tr>
<tr>
<td>1090</td>
<td>NONCRITICAL</td>
</tr>
<tr>
<td>1091</td>
<td>NOAUTHENTIC</td>
</tr>
<tr>
<td>1092</td>
<td>BASICAUTH</td>
</tr>
<tr>
<td>1093</td>
<td>CERTIFICAUTH</td>
</tr>
<tr>
<td>1094</td>
<td>AUTOREGISTER</td>
</tr>
<tr>
<td>1095</td>
<td>AUTOAUTH</td>
</tr>
<tr>
<td>1096</td>
<td>HTTP</td>
</tr>
<tr>
<td>1097</td>
<td>IIOP</td>
</tr>
<tr>
<td>1098</td>
<td>ECI</td>
</tr>
<tr>
<td>1099</td>
<td>VERIFY</td>
</tr>
<tr>
<td>1100</td>
<td>BOTH</td>
</tr>
<tr>
<td>1101</td>
<td>CORBA</td>
</tr>
<tr>
<td>1102</td>
<td>EJB</td>
</tr>
<tr>
<td>1103</td>
<td>HOME</td>
</tr>
<tr>
<td>1104</td>
<td>ASSERTED</td>
</tr>
<tr>
<td>1105</td>
<td>IGNORED</td>
</tr>
<tr>
<td>1106</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>1107</td>
<td>OTS</td>
</tr>
<tr>
<td>Value</td>
<td>CVDA</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>1108</td>
<td>OTSCOORD</td>
</tr>
<tr>
<td>1109</td>
<td>RESPECTED</td>
</tr>
<tr>
<td>1110</td>
<td>NOPHASEOUT</td>
</tr>
<tr>
<td>1111</td>
<td>PHASEOUT</td>
</tr>
<tr>
<td>1112</td>
<td>INITING</td>
</tr>
<tr>
<td>1113</td>
<td>PENDINIT</td>
</tr>
<tr>
<td>1114</td>
<td>PENDRESOLVE</td>
</tr>
<tr>
<td>1115</td>
<td>RESOLVING</td>
</tr>
<tr>
<td>1116</td>
<td>UNRESOLVED</td>
</tr>
<tr>
<td>1117</td>
<td>UNUSABLE</td>
</tr>
<tr>
<td>1118</td>
<td>DISCARDING</td>
</tr>
<tr>
<td>1120</td>
<td>AUTOTERMID</td>
</tr>
<tr>
<td>1121</td>
<td>URMTERMID</td>
</tr>
<tr>
<td>1122</td>
<td>CLIENT</td>
</tr>
<tr>
<td>1123</td>
<td>SERVER</td>
</tr>
<tr>
<td>1124</td>
<td>PIPELINE</td>
</tr>
<tr>
<td>1125</td>
<td>DISABLEDHOST</td>
</tr>
<tr>
<td>1126</td>
<td>ANALYZER</td>
</tr>
<tr>
<td>1127</td>
<td>NOANALYZER</td>
</tr>
<tr>
<td>1140</td>
<td>AUTOINSTALL</td>
</tr>
<tr>
<td>1141</td>
<td>CORBASESERVER</td>
</tr>
<tr>
<td>1142</td>
<td>DB2CONN</td>
</tr>
<tr>
<td>1143</td>
<td>DB2ENTRY</td>
</tr>
<tr>
<td>1144</td>
<td>DISPATCHER</td>
</tr>
<tr>
<td>1145</td>
<td>DOCTEMPLATE</td>
</tr>
<tr>
<td>1146</td>
<td>ENQUEUE</td>
</tr>
<tr>
<td>1147</td>
<td>JOURNALNAME</td>
</tr>
<tr>
<td>1148</td>
<td>JOURNALNUM</td>
</tr>
<tr>
<td>1149</td>
<td>JVMPOOL</td>
</tr>
<tr>
<td>1150</td>
<td>JVMPROFILE</td>
</tr>
<tr>
<td>1151</td>
<td>JVMPROGRAM</td>
</tr>
<tr>
<td>1152</td>
<td>LSRPOOL</td>
</tr>
<tr>
<td>1153</td>
<td>MONITOR</td>
</tr>
<tr>
<td>1154</td>
<td>MVSTCB</td>
</tr>
<tr>
<td>1155</td>
<td>NODE</td>
</tr>
<tr>
<td>1156</td>
<td>RECOVERY</td>
</tr>
<tr>
<td>1157</td>
<td>REQUESTMODEL</td>
</tr>
<tr>
<td>1158</td>
<td>STATS</td>
</tr>
<tr>
<td>1159</td>
<td>STORAGE</td>
</tr>
<tr>
<td>1160</td>
<td>STREAMNAME</td>
</tr>
<tr>
<td>1161</td>
<td>SUBPOOL</td>
</tr>
<tr>
<td>1162</td>
<td>SYSDUMPCODE</td>
</tr>
<tr>
<td>1163</td>
<td>TABLEMGR</td>
</tr>
<tr>
<td>1164</td>
<td>TASKSUBPOOL</td>
</tr>
<tr>
<td>1165</td>
<td>TCPIP</td>
</tr>
<tr>
<td>1166</td>
<td>TCPIPSERVICE</td>
</tr>
<tr>
<td>1167</td>
<td>TDQUEUE</td>
</tr>
<tr>
<td>1168</td>
<td>TEMPLATENAME</td>
</tr>
<tr>
<td>1169</td>
<td>TRANCLASS</td>
</tr>
<tr>
<td>1170</td>
<td>TRANDUMPCODE</td>
</tr>
<tr>
<td>1171</td>
<td>TRANSACTION</td>
</tr>
<tr>
<td>1172</td>
<td>TSQUEUE</td>
</tr>
<tr>
<td>1173</td>
<td>URIMAP</td>
</tr>
<tr>
<td>1174</td>
<td>WEBSERVICE</td>
</tr>
</tbody>
</table>
### CVDA values for the DEVICE option

<table>
<thead>
<tr>
<th>CVDA sequence</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATCHLU</td>
<td>191</td>
</tr>
<tr>
<td>BIPROG</td>
<td>160</td>
</tr>
<tr>
<td>BISYNCH</td>
<td>128</td>
</tr>
<tr>
<td>CDRDLPRT</td>
<td>24</td>
</tr>
<tr>
<td>CONTNLU</td>
<td>189</td>
</tr>
<tr>
<td>HARDCOPY</td>
<td>32</td>
</tr>
<tr>
<td>INTACTLU</td>
<td>190</td>
</tr>
<tr>
<td>ISCMMCONV</td>
<td>209</td>
</tr>
<tr>
<td>LUCMODGRP</td>
<td>210</td>
</tr>
<tr>
<td>LUCSESS</td>
<td>211</td>
</tr>
<tr>
<td>LUTYPE4</td>
<td>193</td>
</tr>
<tr>
<td>LUTYPE6</td>
<td>192</td>
</tr>
<tr>
<td>MAGTAPE</td>
<td>20</td>
</tr>
<tr>
<td>RESSYS</td>
<td>208</td>
</tr>
<tr>
<td>SDLC</td>
<td>176</td>
</tr>
<tr>
<td>SEQDISK</td>
<td>18</td>
</tr>
<tr>
<td>SYSTEM3</td>
<td>161</td>
</tr>
<tr>
<td>SYSTEM7</td>
<td>2</td>
</tr>
<tr>
<td>SYS370</td>
<td>164</td>
</tr>
<tr>
<td>SYS7BSCA</td>
<td>166</td>
</tr>
<tr>
<td>TCONSOLE</td>
<td>8</td>
</tr>
<tr>
<td>TELETYPET</td>
<td>34</td>
</tr>
<tr>
<td>TTCAM</td>
<td>80</td>
</tr>
<tr>
<td>TWX3335</td>
<td>33</td>
</tr>
<tr>
<td>T1050</td>
<td>36</td>
</tr>
<tr>
<td>T1053</td>
<td>74</td>
</tr>
<tr>
<td>T2260L</td>
<td>65</td>
</tr>
<tr>
<td>T2260R</td>
<td>72</td>
</tr>
<tr>
<td>T2265</td>
<td>76</td>
</tr>
<tr>
<td>T2740</td>
<td>40</td>
</tr>
<tr>
<td>T2741BCD</td>
<td>43</td>
</tr>
<tr>
<td>T2741COR</td>
<td>42</td>
</tr>
<tr>
<td>T2770</td>
<td>130</td>
</tr>
<tr>
<td>T2780</td>
<td>132</td>
</tr>
<tr>
<td>T2980</td>
<td>134</td>
</tr>
<tr>
<td>T3275R</td>
<td>146</td>
</tr>
<tr>
<td>T3277L</td>
<td>153</td>
</tr>
<tr>
<td>T3277R</td>
<td>145</td>
</tr>
<tr>
<td>T3284L</td>
<td>155</td>
</tr>
<tr>
<td>T3284R</td>
<td>147</td>
</tr>
<tr>
<td>T3286L</td>
<td>156</td>
</tr>
<tr>
<td>T3286R</td>
<td>148</td>
</tr>
<tr>
<td>T3600BI</td>
<td>138</td>
</tr>
<tr>
<td>T3601</td>
<td>177</td>
</tr>
<tr>
<td>T3614</td>
<td>178</td>
</tr>
<tr>
<td>T3650ATT</td>
<td>186</td>
</tr>
<tr>
<td>T3650PIPE</td>
<td>184</td>
</tr>
<tr>
<td>T3650USER</td>
<td>187</td>
</tr>
<tr>
<td>T3653HOST</td>
<td>185</td>
</tr>
<tr>
<td>T3735</td>
<td>136</td>
</tr>
<tr>
<td>T3740</td>
<td>137</td>
</tr>
<tr>
<td>T3780</td>
<td>133</td>
</tr>
</tbody>
</table>
**CVDA sequence**

- T3790 180
- T3790SCSP 182
- T3790UP 181
- T7770 1
- VIDEOTERM 64

**Numeric sequence**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T7770</td>
</tr>
<tr>
<td>2</td>
<td>SYSTEM7</td>
</tr>
<tr>
<td>8</td>
<td>TCONSOLE</td>
</tr>
<tr>
<td>18</td>
<td>SEQDISK</td>
</tr>
<tr>
<td>20</td>
<td>MAGTAPE</td>
</tr>
<tr>
<td>24</td>
<td>CDRDLPRT</td>
</tr>
<tr>
<td>32</td>
<td>HARDCOPY</td>
</tr>
<tr>
<td>33</td>
<td>TWX3335</td>
</tr>
<tr>
<td>34</td>
<td>TELETYPE</td>
</tr>
<tr>
<td>36</td>
<td>T1050</td>
</tr>
<tr>
<td>40</td>
<td>T2740</td>
</tr>
<tr>
<td>42</td>
<td>T2741COR</td>
</tr>
<tr>
<td>43</td>
<td>T2741BCD</td>
</tr>
<tr>
<td>64</td>
<td>VIDEOTERM</td>
</tr>
<tr>
<td>65</td>
<td>T2260L</td>
</tr>
<tr>
<td>72</td>
<td>T2260R</td>
</tr>
<tr>
<td>74</td>
<td>T1053</td>
</tr>
<tr>
<td>76</td>
<td>T2265</td>
</tr>
<tr>
<td>80</td>
<td>TTCAM</td>
</tr>
<tr>
<td>128</td>
<td>BISYNCH</td>
</tr>
<tr>
<td>130</td>
<td>T2770</td>
</tr>
<tr>
<td>132</td>
<td>T2780</td>
</tr>
<tr>
<td>133</td>
<td>T3780</td>
</tr>
<tr>
<td>134</td>
<td>T2980</td>
</tr>
<tr>
<td>136</td>
<td>T3735</td>
</tr>
<tr>
<td>137</td>
<td>T3740</td>
</tr>
<tr>
<td>138</td>
<td>T3600BI</td>
</tr>
<tr>
<td>145</td>
<td>T3277R</td>
</tr>
<tr>
<td>146</td>
<td>T3275R</td>
</tr>
<tr>
<td>147</td>
<td>T3284R</td>
</tr>
<tr>
<td>148</td>
<td>T3286R</td>
</tr>
<tr>
<td>153</td>
<td>T3277L</td>
</tr>
<tr>
<td>155</td>
<td>T3284L</td>
</tr>
<tr>
<td>156</td>
<td>T3286L</td>
</tr>
<tr>
<td>160</td>
<td>BIPROG</td>
</tr>
<tr>
<td>161</td>
<td>SYSTEM3</td>
</tr>
<tr>
<td>164</td>
<td>SYS370</td>
</tr>
<tr>
<td>166</td>
<td>SYS7BSCA</td>
</tr>
<tr>
<td>176</td>
<td>SDLC</td>
</tr>
<tr>
<td>177</td>
<td>T3601</td>
</tr>
<tr>
<td>178</td>
<td>T3614</td>
</tr>
<tr>
<td>180</td>
<td>T3790</td>
</tr>
<tr>
<td>181</td>
<td>T3790UP</td>
</tr>
<tr>
<td>182</td>
<td>T3790SCSP</td>
</tr>
<tr>
<td>184</td>
<td>T3650PIPE</td>
</tr>
<tr>
<td>185</td>
<td>T3653HOST</td>
</tr>
<tr>
<td>186</td>
<td>T3650ATT</td>
</tr>
</tbody>
</table>

Appendix A. CICS-value data areas used by all commands
Numeric sequence

187  T3650USER
189  CONTNLU
190  INTACTLU
191  BATCHLU
192  LUTYPE6
193  LUTYPE4
208  RESYS
209  ISCMCONV
210  LCUMODGRP
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.

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 the EXEC CICS commands described in this book, their corresponding RESP values (decimal), the associated EIBRCODE values (hexadecimal), and the transaction abend codes (if any).

**EXEC CICS 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'.

<table>
<thead>
<tr>
<th>Condition (Byte 3)</th>
<th>RESP Value code</th>
<th>EIBRCODE</th>
<th>Abend</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNNOTFOUND</td>
<td>93</td>
<td>5D</td>
<td>AEX1</td>
</tr>
<tr>
<td>DUPREC</td>
<td>14</td>
<td>0E</td>
<td>AEIN</td>
</tr>
<tr>
<td>END</td>
<td>83</td>
<td>53</td>
<td>AEXK</td>
</tr>
<tr>
<td>FILENOTFOUND</td>
<td>12</td>
<td>0C</td>
<td>AEIL</td>
</tr>
<tr>
<td>ILLOGIC</td>
<td>21</td>
<td>15</td>
<td>AEIU</td>
</tr>
<tr>
<td>INVREQ</td>
<td>16</td>
<td>10</td>
<td>AEIP</td>
</tr>
<tr>
<td>IOERR</td>
<td>17</td>
<td>11</td>
<td>AEIQ</td>
</tr>
<tr>
<td>JIDERR</td>
<td>43</td>
<td>2B</td>
<td>AEGY</td>
</tr>
<tr>
<td>LENGERR</td>
<td>22</td>
<td>16</td>
<td>AEIV</td>
</tr>
<tr>
<td>MODELIDERR</td>
<td>95</td>
<td>5F</td>
<td>AEX3</td>
</tr>
<tr>
<td>NOSPACE</td>
<td>18</td>
<td>12</td>
<td>AEIR</td>
</tr>
<tr>
<td>NOSTG</td>
<td>42</td>
<td>2A</td>
<td>—</td>
</tr>
<tr>
<td>NOTAUTH</td>
<td>70</td>
<td>46</td>
<td>AEY7</td>
</tr>
<tr>
<td>NOTFND</td>
<td>13</td>
<td>0D</td>
<td>AEIM</td>
</tr>
<tr>
<td>PARTNERIDERR</td>
<td>97</td>
<td>61</td>
<td>AEX5</td>
</tr>
<tr>
<td>PGMIDERR</td>
<td>27</td>
<td>1B</td>
<td>AEI0</td>
</tr>
<tr>
<td>PROFILEIDERR</td>
<td>98</td>
<td>62</td>
<td>AEX6</td>
</tr>
<tr>
<td>QIDERR</td>
<td>44</td>
<td>2C</td>
<td>AEYH</td>
</tr>
<tr>
<td>SYSBUSY</td>
<td>59</td>
<td>3B</td>
<td>—</td>
</tr>
<tr>
<td>SYSIDERR</td>
<td>53</td>
<td>35</td>
<td>AEYQ</td>
</tr>
<tr>
<td>TASKIDERR</td>
<td>91</td>
<td>5B</td>
<td>AEXX</td>
</tr>
<tr>
<td>TOIDERR</td>
<td>92</td>
<td>5C</td>
<td>AEX0</td>
</tr>
<tr>
<td>TERMIDERR</td>
<td>11</td>
<td>0B</td>
<td>AEIK</td>
</tr>
<tr>
<td>TRANSIDERR</td>
<td>28</td>
<td>1C</td>
<td>AEI1</td>
</tr>
<tr>
<td>UOWNOTFOUND</td>
<td>102</td>
<td>66</td>
<td>—</td>
</tr>
<tr>
<td>USERIDERR</td>
<td>69</td>
<td>45</td>
<td>AEYX</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Condition</th>
<th>RESP Value</th>
<th>EIBRCODE</th>
<th>Abend</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVEXITREQ</td>
<td>63</td>
<td>80</td>
<td>AEY0</td>
</tr>
<tr>
<td>NOTAUTH</td>
<td>70</td>
<td>46</td>
<td>AEY7</td>
</tr>
</tbody>
</table>

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. For information about other function codes, see the CICS Application Programming Reference manual.

<table>
<thead>
<tr>
<th>Command</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACQUIRE TERMINAL</td>
<td>86 02</td>
</tr>
<tr>
<td>COLLECT STATISTICS</td>
<td>70 08</td>
</tr>
<tr>
<td>CREATE CONNECTION</td>
<td>30 0E</td>
</tr>
<tr>
<td>CREATE CORBASERVER</td>
<td>30 32</td>
</tr>
<tr>
<td>CREATE DB2CONN</td>
<td>30 20</td>
</tr>
<tr>
<td>CREATE DB2ENTRY</td>
<td>30 22</td>
</tr>
<tr>
<td>CREATE DB2TRAN</td>
<td>30 24</td>
</tr>
<tr>
<td>CREATE DJAR</td>
<td>30 34</td>
</tr>
<tr>
<td>CREATE DOCTEMPLATE</td>
<td>30 2E</td>
</tr>
<tr>
<td>CREATE ENQMODEL</td>
<td>30 2A</td>
</tr>
<tr>
<td>CREATE FILE</td>
<td>30 14</td>
</tr>
<tr>
<td>CREATE JOURNALMODEL</td>
<td>30 1E</td>
</tr>
<tr>
<td>CREATE LSRPOOL</td>
<td>30 16</td>
</tr>
<tr>
<td>CREATE MAPSET</td>
<td>30 04</td>
</tr>
<tr>
<td>CREATE PARTITIONSET</td>
<td>30 06</td>
</tr>
<tr>
<td>CREATE PARTNER</td>
<td>30 18</td>
</tr>
<tr>
<td>CREATE PIPELINE</td>
<td>30 38</td>
</tr>
<tr>
<td>CREATE PROCESSTYPE</td>
<td>30 26</td>
</tr>
<tr>
<td>CREATE PROFILE</td>
<td>30 0A</td>
</tr>
<tr>
<td>CREATE PROGRAM</td>
<td>30 02</td>
</tr>
<tr>
<td>CREATE REQUESTMODEL</td>
<td>30 2C</td>
</tr>
<tr>
<td>CREATE SESSIONS</td>
<td>30 12</td>
</tr>
<tr>
<td>CREATE TCPIPSERVICE</td>
<td>30 30</td>
</tr>
<tr>
<td>CREATE TDQUEUE</td>
<td>30 1C</td>
</tr>
<tr>
<td>CREATE TERMINAL</td>
<td>30 10</td>
</tr>
<tr>
<td>CREATE TRANCLASS</td>
<td>30 1A</td>
</tr>
<tr>
<td>CREATE TRANSACTION</td>
<td>30 08</td>
</tr>
<tr>
<td>Command</td>
<td>Code</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>CREATE TSMODEL</td>
<td>30 28</td>
</tr>
<tr>
<td>CREATE TYPETERM</td>
<td>30 0C</td>
</tr>
<tr>
<td>CREATE URIMAP</td>
<td>30 36</td>
</tr>
<tr>
<td>CREATE WEBSERVICE</td>
<td>30 3A</td>
</tr>
<tr>
<td>DISABLE PROGRAM</td>
<td>22 04</td>
</tr>
<tr>
<td>DISCARD AUTINSTMODEL</td>
<td>42 10</td>
</tr>
<tr>
<td>DISCARD CONNECTION</td>
<td>58 10</td>
</tr>
<tr>
<td>DISCARD CORBASERVER</td>
<td>B2 10</td>
</tr>
<tr>
<td>DISCARD DB2CONN</td>
<td>94 10</td>
</tr>
<tr>
<td>DISCARD DB2ENTRY</td>
<td>94 30</td>
</tr>
<tr>
<td>DISCARD DB2TRAN</td>
<td>94 50</td>
</tr>
<tr>
<td>DISCARD DJAR</td>
<td>B2 30</td>
</tr>
<tr>
<td>DISCARD DOCTEMPLATE</td>
<td>9E 10</td>
</tr>
<tr>
<td>DISCARD ENQMODEL</td>
<td>90 90</td>
</tr>
<tr>
<td>DISCARD FILE</td>
<td>4C 10</td>
</tr>
<tr>
<td>DISCARD JOURNALMODEL</td>
<td>92 10</td>
</tr>
<tr>
<td>DISCARD JOURNALNAME</td>
<td>60 10</td>
</tr>
<tr>
<td>DISCARD PARTNER</td>
<td>44 10</td>
</tr>
<tr>
<td>DISCARD PIPELINE</td>
<td>BC 10</td>
</tr>
<tr>
<td>DISCARD PROCESTYPE</td>
<td>96 10</td>
</tr>
<tr>
<td>DISCARD PROFILE</td>
<td>46 10</td>
</tr>
<tr>
<td>DISCARD PROGRAM</td>
<td>4E 10</td>
</tr>
<tr>
<td>DISCARD REQUESTMODEL</td>
<td>9A 10</td>
</tr>
<tr>
<td>DISCARD TCPIPSERVICE</td>
<td>9C 10</td>
</tr>
<tr>
<td>DISCARD TDQUEUE</td>
<td>5C 10</td>
</tr>
<tr>
<td>DISCARD TERMINAL</td>
<td>52 10</td>
</tr>
<tr>
<td>DISCARD TRANCLASS</td>
<td>5E 18</td>
</tr>
<tr>
<td>DISCARD TRANSACTION</td>
<td>50 10</td>
</tr>
<tr>
<td>DISCARD TSMODEL</td>
<td>80 30</td>
</tr>
<tr>
<td>DISCARD URIMAP</td>
<td>BE 10</td>
</tr>
<tr>
<td>DISCARD WEBSERVICE</td>
<td>BC 30</td>
</tr>
<tr>
<td>ENABLE PROGRAM</td>
<td>22 02</td>
</tr>
<tr>
<td>EXTRACT EXIT</td>
<td>22 06</td>
</tr>
<tr>
<td>EXTRACT STATISTICS</td>
<td>70 26</td>
</tr>
<tr>
<td>INQUIRE AUTINSTMODEL</td>
<td>42 02</td>
</tr>
<tr>
<td>INQUIRE AUTOINSTALL</td>
<td>68 12</td>
</tr>
<tr>
<td>INQUIRE BEAN</td>
<td>B2 42</td>
</tr>
<tr>
<td>INQUIRE BRFACILITY</td>
<td>B4 02</td>
</tr>
<tr>
<td>INQUIRE CFDTPOOL</td>
<td>98 02</td>
</tr>
<tr>
<td>INQUIRE CLASSCACHE</td>
<td>B0 22</td>
</tr>
<tr>
<td>INQUIRE CONNECTION</td>
<td>58 02</td>
</tr>
<tr>
<td>INQUIRE CORBASERVER</td>
<td>B2 02</td>
</tr>
<tr>
<td>INQUIRE DB2CONN</td>
<td>94 02</td>
</tr>
<tr>
<td>INQUIRE DB2ENTRY</td>
<td>94 22</td>
</tr>
<tr>
<td>INQUIRE DB2TRAN</td>
<td>94 42</td>
</tr>
<tr>
<td>INQUIRE DELETESHIPPED</td>
<td>68 22</td>
</tr>
<tr>
<td>INQUIRE DISPATCHER</td>
<td>B6 02</td>
</tr>
<tr>
<td>INQUIRE DJAR</td>
<td>B2 22</td>
</tr>
<tr>
<td>INQUIRE DOCTEMPLATE</td>
<td>9E 02</td>
</tr>
<tr>
<td>INQUIRE DSNNAME</td>
<td>7A 02</td>
</tr>
<tr>
<td>INQUIRE DUMPDS</td>
<td>66 02</td>
</tr>
<tr>
<td>INQUIRE ENQ</td>
<td>90 22</td>
</tr>
<tr>
<td>INQUIRE ENQMODEL</td>
<td>90 82</td>
</tr>
<tr>
<td>INQUIRE EXCI</td>
<td>7C 02</td>
</tr>
<tr>
<td>Command</td>
<td>Code</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>INQUIRE EXITPROGRAM</td>
<td>88 02</td>
</tr>
<tr>
<td>INQUIRE FILE</td>
<td>4C 02</td>
</tr>
<tr>
<td>INQUIRE HOST</td>
<td>BE 12</td>
</tr>
<tr>
<td>INQUIRE IRC</td>
<td>6E 02</td>
</tr>
<tr>
<td>INQUIRE JOURNALMODEL</td>
<td>92 02</td>
</tr>
<tr>
<td>INQUIRE JOURNALNAME</td>
<td>60 12</td>
</tr>
<tr>
<td>INQUIRE JOURNALNUM</td>
<td>60 02</td>
</tr>
<tr>
<td>INQUIRE JVM</td>
<td>B0 32</td>
</tr>
<tr>
<td>INQUIRE JVMPOOL</td>
<td>B0 02</td>
</tr>
<tr>
<td>INQUIRE JVMPROFILE</td>
<td>B0 12</td>
</tr>
<tr>
<td>INQUIRE MODENAME</td>
<td>5A 02</td>
</tr>
<tr>
<td>INQUIRE MONITOR</td>
<td>70 12</td>
</tr>
<tr>
<td>INQUIRE MVSTCB</td>
<td>B6 12</td>
</tr>
<tr>
<td>INQUIRE NETNAME</td>
<td>52 16</td>
</tr>
<tr>
<td>INQUIRE NETNAME</td>
<td>52 06</td>
</tr>
<tr>
<td>INQUIRE PARTNER</td>
<td>44 02</td>
</tr>
<tr>
<td>INQUIRE PIPELINE</td>
<td>BC 02</td>
</tr>
<tr>
<td>INQUIRE PROCESSTYPE</td>
<td>96 02</td>
</tr>
<tr>
<td>INQUIRE PROFILE</td>
<td>46 02</td>
</tr>
<tr>
<td>INQUIRE PROGRAM</td>
<td>4E 02</td>
</tr>
<tr>
<td>INQUIRE REOID</td>
<td>8A 02</td>
</tr>
<tr>
<td>INQUIRE REQUESTMODEL</td>
<td>9A 02</td>
</tr>
<tr>
<td>INQUIRE RRMS</td>
<td>3A 02</td>
</tr>
<tr>
<td>INQUIRE STATISTICS</td>
<td>70 02</td>
</tr>
<tr>
<td>INQUIRE STORAGE</td>
<td>5E 08</td>
</tr>
<tr>
<td>INQUIRE STREAMNAME</td>
<td>92 12</td>
</tr>
<tr>
<td>INQUIRE SUBPOOL</td>
<td>5E 42</td>
</tr>
<tr>
<td>INQUIRE SYSDUMPCODE</td>
<td>66 22</td>
</tr>
<tr>
<td>INQUIRE SYSTEM</td>
<td>54 02</td>
</tr>
<tr>
<td>INQUIRE TASK</td>
<td>5E 02</td>
</tr>
<tr>
<td>INQUIRE TCLASS</td>
<td>5E 12</td>
</tr>
<tr>
<td>INQUIRE TCPIP</td>
<td>9C 12</td>
</tr>
<tr>
<td>INQUIRE TCPIPSERVICE</td>
<td>9C 02</td>
</tr>
<tr>
<td>INQUIRE TDQUEUE</td>
<td>5C 02</td>
</tr>
<tr>
<td>INQUIRE TERMINAL</td>
<td>52 02</td>
</tr>
<tr>
<td>INQUIRE TERMINAL</td>
<td>52 12</td>
</tr>
<tr>
<td>INQUIRE TRACEDEST</td>
<td>78 02</td>
</tr>
<tr>
<td>INQUIRE TRACEFLAG</td>
<td>78 12</td>
</tr>
<tr>
<td>INQUIRE TRACETYPE</td>
<td>78 22</td>
</tr>
<tr>
<td>INQUIRE TRANCLASS</td>
<td>5E 1A</td>
</tr>
<tr>
<td>INQUIRE TRANDUMPCODE</td>
<td>5E 12</td>
</tr>
<tr>
<td>INQUIRE TRANSACTION</td>
<td>50 02</td>
</tr>
<tr>
<td>INQUIRE TMODEL</td>
<td>80 22</td>
</tr>
<tr>
<td>INQUIRE TPOOL</td>
<td>80 1A</td>
</tr>
<tr>
<td>INQUIRE TSQNAME</td>
<td>80 12</td>
</tr>
<tr>
<td>INQUIRE TSQUEUE</td>
<td>80 02</td>
</tr>
<tr>
<td>INQUIRE UOW</td>
<td>90 02</td>
</tr>
<tr>
<td>INQUIRE UOWDSNFAIL</td>
<td>90 62</td>
</tr>
<tr>
<td>INQUIRE UOWENQ</td>
<td>90 22</td>
</tr>
<tr>
<td>INQUIRE UOWLINK</td>
<td>90 42</td>
</tr>
<tr>
<td>INQUIRE UURIMAP</td>
<td>BE 02</td>
</tr>
<tr>
<td>INQUIRE UVOLUME</td>
<td>62 02</td>
</tr>
<tr>
<td>INQUIRE VTAM</td>
<td>68 02</td>
</tr>
<tr>
<td>INQUIRE WEB</td>
<td>9C 22</td>
</tr>
<tr>
<td>Command</td>
<td>Code</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>INQUIRE WEBSERVICE</td>
<td>BC 22</td>
</tr>
<tr>
<td>INQUIRE WORKREQUEST</td>
<td>B8 02</td>
</tr>
<tr>
<td>PERFORM CLASSCACHE</td>
<td>B0 26</td>
</tr>
<tr>
<td>PERFORM CORBASERVER</td>
<td>B2 06</td>
</tr>
<tr>
<td>PERFORM DELETISHED</td>
<td>68 26</td>
</tr>
<tr>
<td>PERFORM DJAR</td>
<td>B2 24</td>
</tr>
<tr>
<td>PERFORM DUMP</td>
<td>7E 04</td>
</tr>
<tr>
<td>PERFORM ENDAFFINITY</td>
<td>58 06</td>
</tr>
<tr>
<td>PERFORM PIPELINE</td>
<td>BC 06</td>
</tr>
<tr>
<td>PERFORM RESETTIME</td>
<td>72 02</td>
</tr>
<tr>
<td>PERFORM SECURITY</td>
<td>64 02</td>
</tr>
<tr>
<td>PERFORM SHUTDOWN</td>
<td>76 02</td>
</tr>
<tr>
<td>PERFORM STATISTICS</td>
<td>70 06</td>
</tr>
<tr>
<td>RESYNC ENTRYNAMES</td>
<td>16 04</td>
</tr>
<tr>
<td>SET AUTOINSTALL</td>
<td>68 14</td>
</tr>
<tr>
<td>SET BRFACILITY</td>
<td>B4 04</td>
</tr>
<tr>
<td>SET CLASSCACHE</td>
<td>B0 24</td>
</tr>
<tr>
<td>SET CONNECTION</td>
<td>58 04</td>
</tr>
<tr>
<td>SET CORBASERVER</td>
<td>B2 04</td>
</tr>
<tr>
<td>SET DB2CONN</td>
<td>94 04</td>
</tr>
<tr>
<td>SET DB2ENTRY</td>
<td>94 24</td>
</tr>
<tr>
<td>SET DB2TRAN</td>
<td>94 44</td>
</tr>
<tr>
<td>SET DELETISHED</td>
<td>68 24</td>
</tr>
<tr>
<td>SET DISPATCHER</td>
<td>B6 04</td>
</tr>
<tr>
<td>SET DSNAMEN</td>
<td>7A 04</td>
</tr>
<tr>
<td>SET DUMPDS</td>
<td>66 04</td>
</tr>
<tr>
<td>SET ENQMODEL</td>
<td>90 84</td>
</tr>
<tr>
<td>SET FILE</td>
<td>4C 04</td>
</tr>
<tr>
<td>SET HOST</td>
<td>BE 14</td>
</tr>
<tr>
<td>SET IRC</td>
<td>6E 04</td>
</tr>
<tr>
<td>SET JOURNALNAME</td>
<td>60 14</td>
</tr>
<tr>
<td>SET JOURNALNUM</td>
<td>60 04</td>
</tr>
<tr>
<td>SET JVMPOOL</td>
<td>B0 04</td>
</tr>
<tr>
<td>SET MODENAME</td>
<td>5A 04</td>
</tr>
<tr>
<td>SET MONITOR</td>
<td>70 14</td>
</tr>
<tr>
<td>SET NETNAME</td>
<td>52 08</td>
</tr>
<tr>
<td>SET PIPELINE</td>
<td>BC 04</td>
</tr>
<tr>
<td>SET PROCESSTYPE</td>
<td>96 04</td>
</tr>
<tr>
<td>SET PROGRAM</td>
<td>4E 04</td>
</tr>
<tr>
<td>SET STATISTICS</td>
<td>70 04</td>
</tr>
<tr>
<td>SET SYSDUMPCODE</td>
<td>66 24</td>
</tr>
<tr>
<td>SET SYSTEM</td>
<td>54 04</td>
</tr>
<tr>
<td>SET TASK</td>
<td>5E 04</td>
</tr>
<tr>
<td>SET TCLASS</td>
<td>5E 14</td>
</tr>
<tr>
<td>SET TCPIP</td>
<td>9C 14</td>
</tr>
<tr>
<td>SET TCPIPSERVICE</td>
<td>9C 04</td>
</tr>
<tr>
<td>SET TDQUEUE</td>
<td>5C 04</td>
</tr>
<tr>
<td>SET TERMINAL</td>
<td>52 04</td>
</tr>
<tr>
<td>SET TERMINAL</td>
<td>52 14</td>
</tr>
<tr>
<td>SET TRACEDEST</td>
<td>78 04</td>
</tr>
<tr>
<td>SET TRACEDEST</td>
<td>78 14</td>
</tr>
<tr>
<td>SET TRACETYPE</td>
<td>78 24</td>
</tr>
<tr>
<td>SET TRANCLASS</td>
<td>5E 1C</td>
</tr>
<tr>
<td>SET TRANDUMPCODE</td>
<td>66 14</td>
</tr>
</tbody>
</table>
### Command Code

<table>
<thead>
<tr>
<th>Command</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET TRANSACTION</td>
<td>50 04</td>
</tr>
<tr>
<td>SET TSONAME</td>
<td>80 14</td>
</tr>
<tr>
<td>SET TSQUEUE</td>
<td>80 04</td>
</tr>
<tr>
<td>SET UOW</td>
<td>90 04</td>
</tr>
<tr>
<td>SET UOWLINK</td>
<td>90 44</td>
</tr>
<tr>
<td>SET URIMAP</td>
<td>BE 04</td>
</tr>
<tr>
<td>SET VOLUME</td>
<td>62 04</td>
</tr>
<tr>
<td>SET VTAM</td>
<td>68 04</td>
</tr>
<tr>
<td>SET WEB</td>
<td>9C 24</td>
</tr>
<tr>
<td>SET WEBSERVICE</td>
<td>BC 24</td>
</tr>
<tr>
<td>SET WORKREQUEST</td>
<td>B8 04</td>
</tr>
</tbody>
</table>

### Code Command

<table>
<thead>
<tr>
<th>Code</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 04</td>
<td>RESYNC ENTRYNAME</td>
</tr>
<tr>
<td>22 02</td>
<td>ENABLE PROGRAM</td>
</tr>
<tr>
<td>22 04</td>
<td>DISABLE PROGRAM</td>
</tr>
<tr>
<td>22 06</td>
<td>EXTRACT EXIT</td>
</tr>
<tr>
<td>30 0A</td>
<td>CREATE PROFILE</td>
</tr>
<tr>
<td>30 0C</td>
<td>CREATE TYPETERM</td>
</tr>
<tr>
<td>30 0E</td>
<td>CREATE CONNECTION</td>
</tr>
<tr>
<td>30 02</td>
<td>CREATE PROGRAM</td>
</tr>
<tr>
<td>30 04</td>
<td>CREATE MAPSET</td>
</tr>
<tr>
<td>30 06</td>
<td>CREATE PARTITIONSET</td>
</tr>
<tr>
<td>30 08</td>
<td>CREATE TRANSACTION</td>
</tr>
<tr>
<td>30 1A</td>
<td>CREATE TRANCLASS</td>
</tr>
<tr>
<td>30 1C</td>
<td>CREATE TDQUEUE</td>
</tr>
<tr>
<td>30 1E</td>
<td>CREATE JOURNALMODEL</td>
</tr>
<tr>
<td>30 10</td>
<td>CREATE TERMINAL</td>
</tr>
<tr>
<td>30 12</td>
<td>CREATE SESSIONS</td>
</tr>
<tr>
<td>30 14</td>
<td>CREATE FILE</td>
</tr>
<tr>
<td>30 16</td>
<td>CREATE LSRPOOL</td>
</tr>
<tr>
<td>30 18</td>
<td>CREATE PARTNER</td>
</tr>
<tr>
<td>30 20</td>
<td>CREATE DB2CONN</td>
</tr>
<tr>
<td>30 22</td>
<td>CREATE DB2ENTRY</td>
</tr>
<tr>
<td>30 24</td>
<td>CREATE DB2TRAN</td>
</tr>
<tr>
<td>30 26</td>
<td>CREATE PROCESSTYPE</td>
</tr>
<tr>
<td>30 28</td>
<td>CREATE TSMODEL</td>
</tr>
<tr>
<td>30 2A</td>
<td>CREATE ENQMODEL</td>
</tr>
<tr>
<td>30 2C</td>
<td>CREATE REQUESTMODEL</td>
</tr>
<tr>
<td>30 2E</td>
<td>CREATE DOCTEMPLATE</td>
</tr>
<tr>
<td>30 30</td>
<td>CREATE TCPIPSERVICE</td>
</tr>
<tr>
<td>30 32</td>
<td>CREATE CORBASERVER</td>
</tr>
<tr>
<td>30 34</td>
<td>CREATE DJAR</td>
</tr>
<tr>
<td>#</td>
<td></td>
</tr>
<tr>
<td>30 36</td>
<td>CREATE URIMAP</td>
</tr>
<tr>
<td>#</td>
<td></td>
</tr>
<tr>
<td>30 38</td>
<td>CREATE PIPELINE</td>
</tr>
<tr>
<td>#</td>
<td></td>
</tr>
<tr>
<td>30 3A</td>
<td>CREATE WEBSERVICE</td>
</tr>
<tr>
<td>3A 02</td>
<td>INQUIRE RRMS</td>
</tr>
<tr>
<td>42 02</td>
<td>INQUIRE AUTINSTMODEL</td>
</tr>
<tr>
<td>42 10</td>
<td>DISCARD AUTINSTMODEL</td>
</tr>
<tr>
<td>44 02</td>
<td>INQUIRE PARTNER</td>
</tr>
<tr>
<td>44 10</td>
<td>DISCARD PARTNER</td>
</tr>
<tr>
<td>46 02</td>
<td>INQUIRE PROFILE</td>
</tr>
<tr>
<td>Code</td>
<td>Command</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>46 10</td>
<td>DISCARD PROFILE</td>
</tr>
<tr>
<td>4C 02</td>
<td>INQUIRE FILE</td>
</tr>
<tr>
<td>4C 04</td>
<td>SET FILE</td>
</tr>
<tr>
<td>4C 10</td>
<td>DISCARD FILE</td>
</tr>
<tr>
<td>4E 02</td>
<td>INQUIRE PROGRAM</td>
</tr>
<tr>
<td>4E 04</td>
<td>SET PROGRAM</td>
</tr>
<tr>
<td>4E 10</td>
<td>DISCARD PROGRAM</td>
</tr>
<tr>
<td>50 02</td>
<td>INQUIRE TRANSACTION</td>
</tr>
<tr>
<td>50 04</td>
<td>SET TRANSACTION</td>
</tr>
<tr>
<td>50 10</td>
<td>DISCARD TRANSACTION</td>
</tr>
<tr>
<td>52 02</td>
<td>INQUIRE TERMINAL</td>
</tr>
<tr>
<td>52 04</td>
<td>SET TERMINAL</td>
</tr>
<tr>
<td>52 06</td>
<td>INQUIRE NETNAME</td>
</tr>
<tr>
<td>52 08</td>
<td>SET NETNAME</td>
</tr>
<tr>
<td>52 10</td>
<td>DISCARD TERMINAL</td>
</tr>
<tr>
<td>52 12</td>
<td>INQUIRE TERMINAL</td>
</tr>
<tr>
<td>52 14</td>
<td>SET TERMINAL</td>
</tr>
<tr>
<td>52 16</td>
<td>INQUIRE NETNAME</td>
</tr>
<tr>
<td>54 02</td>
<td>INQUIRE SYSTEM</td>
</tr>
<tr>
<td>54 04</td>
<td>SET SYSTEM</td>
</tr>
<tr>
<td>58 02</td>
<td>INQUIRE CONNECTION</td>
</tr>
<tr>
<td>58 04</td>
<td>SET CONNECTION</td>
</tr>
<tr>
<td>58 06</td>
<td>PERFORM ENDAFFINITY</td>
</tr>
<tr>
<td>58 10</td>
<td>DISCARD CONNECTION</td>
</tr>
<tr>
<td>5A 02</td>
<td>INQUIRE MODENAME</td>
</tr>
<tr>
<td>5A 04</td>
<td>SET MODENAME</td>
</tr>
<tr>
<td>5C 02</td>
<td>INQUIRE TDQUEUE</td>
</tr>
<tr>
<td>5C 04</td>
<td>SET TDQUEUE</td>
</tr>
<tr>
<td>5C 10</td>
<td>DISCARD TDQUEUE</td>
</tr>
<tr>
<td>5E 02</td>
<td>INQUIRE TASK</td>
</tr>
<tr>
<td>5E 04</td>
<td>SET TASK</td>
</tr>
<tr>
<td>5E 08</td>
<td>INQUIRE STORAGE</td>
</tr>
<tr>
<td>5E 1A</td>
<td>INQUIRE TRANCLASS</td>
</tr>
<tr>
<td>5E 1C</td>
<td>SET TRANCLASS</td>
</tr>
<tr>
<td>5E 12</td>
<td>INQUIRE TCLASS</td>
</tr>
<tr>
<td>5E 14</td>
<td>SET TCLASS</td>
</tr>
<tr>
<td>5E 18</td>
<td>DISCARD TRANCLASS</td>
</tr>
<tr>
<td>5E 42</td>
<td>INQUIRE SUBPOOL</td>
</tr>
<tr>
<td>60 02</td>
<td>INQUIRE JOURNALNUM</td>
</tr>
<tr>
<td>60 04</td>
<td>SET JOURNALNUM</td>
</tr>
<tr>
<td>60 10</td>
<td>DISCARD JOURNALNAME</td>
</tr>
<tr>
<td>60 12</td>
<td>INQUIRE JOURNALNAME</td>
</tr>
<tr>
<td>60 14</td>
<td>SET JOURNALNAME</td>
</tr>
<tr>
<td>62 02</td>
<td>INQUIRE VOLUME</td>
</tr>
<tr>
<td>62 04</td>
<td>SET VOLUME</td>
</tr>
<tr>
<td>64 02</td>
<td>PERFORM SECURITY</td>
</tr>
<tr>
<td>66 02</td>
<td>INQUIRE DUMPPDS</td>
</tr>
<tr>
<td>66 04</td>
<td>SET DUMPPDS</td>
</tr>
<tr>
<td>66 12</td>
<td>INQUIRE TRANSDUMPCODE</td>
</tr>
<tr>
<td>66 14</td>
<td>SET TRANSDUMPCODE</td>
</tr>
<tr>
<td>66 22</td>
<td>INQUIRE SYSDUMPCODE</td>
</tr>
<tr>
<td>66 24</td>
<td>SET SYSDUMPCODE</td>
</tr>
<tr>
<td>68 02</td>
<td>INQUIRE VTAM</td>
</tr>
<tr>
<td>68 04</td>
<td>SET VTAM</td>
</tr>
</tbody>
</table>

Appendix B. EXEC interface block (EIB) response and function codes  677
<table>
<thead>
<tr>
<th>Code</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>68 12</td>
<td>INQUIRE AUTOINSTALL</td>
</tr>
<tr>
<td>68 14</td>
<td>SET AUTOINSTALL</td>
</tr>
<tr>
<td>68 22</td>
<td>INQUIRE DELETESHIPPED</td>
</tr>
<tr>
<td>68 24</td>
<td>SET DELETESHIPPED</td>
</tr>
<tr>
<td>68 26</td>
<td>PERFORM DELETESHIPPED</td>
</tr>
<tr>
<td>6E 02</td>
<td>INQUIRE IRC</td>
</tr>
<tr>
<td>6E 04</td>
<td>SET IRC</td>
</tr>
<tr>
<td>70 02</td>
<td>INQUIRE STATISTICS</td>
</tr>
<tr>
<td>70 04</td>
<td>SET STATISTICS</td>
</tr>
<tr>
<td>70 06</td>
<td>PERFORM STATISTICS</td>
</tr>
<tr>
<td>70 08</td>
<td>COLLECT STATISTICS</td>
</tr>
<tr>
<td>70 12</td>
<td>INQUIRE MONITOR</td>
</tr>
<tr>
<td>70 14</td>
<td>SET MONITOR</td>
</tr>
<tr>
<td>70 26</td>
<td>EXTRACT STATISTICS</td>
</tr>
<tr>
<td>72 02</td>
<td>PERFORM RESETTIME</td>
</tr>
<tr>
<td>76 02</td>
<td>PERFORM SHUTDOWN</td>
</tr>
<tr>
<td>78 02</td>
<td>INQUIRE TRACEDEST</td>
</tr>
<tr>
<td>78 04</td>
<td>SET TRACEDEST</td>
</tr>
<tr>
<td>78 12</td>
<td>INQUIRE TRACEFLAG</td>
</tr>
<tr>
<td>78 14</td>
<td>SET TRACEFLAG</td>
</tr>
<tr>
<td>78 22</td>
<td>INQUIRE TRACETYPE</td>
</tr>
<tr>
<td>78 24</td>
<td>SET TRACETYPE</td>
</tr>
<tr>
<td>7A 02</td>
<td>INQUIRE DNAME</td>
</tr>
<tr>
<td>7A 04</td>
<td>SET DNAME</td>
</tr>
<tr>
<td>7C 02</td>
<td>INQUIRE EXCI</td>
</tr>
<tr>
<td>7E 04</td>
<td>PERFORM DUMP</td>
</tr>
<tr>
<td>80 02</td>
<td>INQUIRE TSQUEUE</td>
</tr>
<tr>
<td>80 04</td>
<td>SET TSQUEUE</td>
</tr>
<tr>
<td>80 12</td>
<td>INQUIRE TSONAME</td>
</tr>
<tr>
<td>80 14</td>
<td>SET TSONAME</td>
</tr>
<tr>
<td>80 1A</td>
<td>INQUIRE TSPool</td>
</tr>
<tr>
<td>80 22</td>
<td>INQUIRE TSMODEL</td>
</tr>
<tr>
<td>80 30</td>
<td>DISCARD TSMODEL</td>
</tr>
<tr>
<td>86 02</td>
<td>ACQUIRE TERMINAL</td>
</tr>
<tr>
<td>88 02</td>
<td>INQUIRE EXITPROGRAM</td>
</tr>
<tr>
<td>8A 02</td>
<td>INQUIRE REQID</td>
</tr>
<tr>
<td>90 02</td>
<td>INQUIRE UOW</td>
</tr>
<tr>
<td>90 04</td>
<td>SET UOW</td>
</tr>
<tr>
<td>90 22</td>
<td>INQUIRE ENQ</td>
</tr>
<tr>
<td>90 22</td>
<td>INQUIRE UOWENQ</td>
</tr>
<tr>
<td>90 42</td>
<td>INQUIRE UOWLINK</td>
</tr>
<tr>
<td>90 44</td>
<td>SET UOWLINK</td>
</tr>
<tr>
<td>90 62</td>
<td>INQUIRE UOWDSNFAIL</td>
</tr>
<tr>
<td>90 82</td>
<td>INQUIRE ENQMODEL</td>
</tr>
<tr>
<td>90 84</td>
<td>SET ENQMODEL</td>
</tr>
<tr>
<td>90 90</td>
<td>DISCARD ENQMODEL</td>
</tr>
<tr>
<td>92 02</td>
<td>INQUIRE JOURNALMODEL</td>
</tr>
<tr>
<td>92 10</td>
<td>DISCARD JOURNALMODEL</td>
</tr>
<tr>
<td>92 12</td>
<td>INQUIRE STREAMNAME</td>
</tr>
<tr>
<td>94 02</td>
<td>INQUIRE DB2CONN</td>
</tr>
<tr>
<td>94 04</td>
<td>SET DB2CONN</td>
</tr>
<tr>
<td>94 10</td>
<td>DISCARD DB2CONN</td>
</tr>
<tr>
<td>94 22</td>
<td>INQUIRE DB2ENTRY</td>
</tr>
<tr>
<td>94 24</td>
<td>SET DB2ENTRY</td>
</tr>
<tr>
<td>Code</td>
<td>Command</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>94 30</td>
<td>DISCARD DB2ENTRY</td>
</tr>
<tr>
<td>94 42</td>
<td>INQUIRE DB2TRAN</td>
</tr>
<tr>
<td>94 44</td>
<td>SET DB2TRAN</td>
</tr>
<tr>
<td>94 50</td>
<td>DISCARD DB2TRAN</td>
</tr>
<tr>
<td>96 02</td>
<td>INQUIRE PROCESSTYPE</td>
</tr>
<tr>
<td>96 04</td>
<td>SET PROCESSTYPE</td>
</tr>
<tr>
<td>96 10</td>
<td>DISCARD PROCESSTYPE</td>
</tr>
<tr>
<td>96 02</td>
<td>INQUIRE CFDTPOOL</td>
</tr>
<tr>
<td>9A 02</td>
<td>DISCARD REQUESTMODEL</td>
</tr>
<tr>
<td>9A 10</td>
<td>DISCARD REQUESTMODEL</td>
</tr>
<tr>
<td>9C 02</td>
<td>INQUIRE TCPIPSERVICE</td>
</tr>
<tr>
<td>9C 04</td>
<td>SET TCPIPSERVICE</td>
</tr>
<tr>
<td>9C 10</td>
<td>DISCARD TCPIPSERVICE</td>
</tr>
<tr>
<td>9C 12</td>
<td>INQUIRE TCPIP</td>
</tr>
<tr>
<td>9C 14</td>
<td>SET TCPIP</td>
</tr>
<tr>
<td>9C 22</td>
<td>INQUIRE WEB</td>
</tr>
<tr>
<td>9C 24</td>
<td>SET WEB</td>
</tr>
<tr>
<td>9E 02</td>
<td>INQUIRE DOCTEMPLATE</td>
</tr>
<tr>
<td>9E 10</td>
<td>DISCARD DOCTEMPLATE</td>
</tr>
<tr>
<td>B0 02</td>
<td>INQUIRE JVMPOOL</td>
</tr>
<tr>
<td>B0 04</td>
<td>SET JVMPOOL</td>
</tr>
<tr>
<td>B0 12</td>
<td>INQUIRE JVMPROFILE</td>
</tr>
<tr>
<td>B0 22</td>
<td>INQUIRE CLASSCACHE</td>
</tr>
<tr>
<td>B0 24</td>
<td>SET CLASSCACHE</td>
</tr>
<tr>
<td>B0 26</td>
<td>PERFORM CLASSCACHE</td>
</tr>
<tr>
<td>B0 32</td>
<td>INQUIRE JVM</td>
</tr>
<tr>
<td>B2 02</td>
<td>INQUIRE CORBASERVER</td>
</tr>
<tr>
<td>B2 04</td>
<td>SET CORBASERVER</td>
</tr>
<tr>
<td>B2 06</td>
<td>PERFORM CORBASERVER</td>
</tr>
<tr>
<td>B2 10</td>
<td>DISCARD CORBASERVER</td>
</tr>
<tr>
<td>B2 22</td>
<td>INQUIRE DJAR</td>
</tr>
<tr>
<td>B2 24</td>
<td>PERFORM DJAR</td>
</tr>
<tr>
<td>B2 30</td>
<td>DISCARD DJAR</td>
</tr>
<tr>
<td>B2 42</td>
<td>INQUIRE BEAN</td>
</tr>
<tr>
<td>B4 02</td>
<td>INQUIRE BRFACILITY</td>
</tr>
<tr>
<td>B4 04</td>
<td>SET BRFACILITY</td>
</tr>
<tr>
<td>B6 02</td>
<td>INQUIRE DISPATCHER</td>
</tr>
<tr>
<td>B6 04</td>
<td>SET DISPATCHER</td>
</tr>
<tr>
<td>B6 12</td>
<td>INQUIRE MVSTCB</td>
</tr>
<tr>
<td>B8 02</td>
<td>INQUIRE WORKREQUEST</td>
</tr>
<tr>
<td>B8 04</td>
<td>SET WORKREQUEST</td>
</tr>
<tr>
<td>#</td>
<td>INQUIRE PIPELINE</td>
</tr>
<tr>
<td>#</td>
<td>SET PIPELINE</td>
</tr>
<tr>
<td>#</td>
<td>PERFORM PIPELINE</td>
</tr>
<tr>
<td>#</td>
<td>DISCARD PIPELINE</td>
</tr>
<tr>
<td>#</td>
<td>INQUIRE WEBSERVICE</td>
</tr>
<tr>
<td>#</td>
<td>SET WEBSERVICE</td>
</tr>
<tr>
<td>#</td>
<td>DISCARD WEBSERVICE</td>
</tr>
<tr>
<td>#</td>
<td>INQUIRE URIMAP</td>
</tr>
<tr>
<td>#</td>
<td>SET URIMAP</td>
</tr>
<tr>
<td>#</td>
<td>DISCARD URIMAP</td>
</tr>
<tr>
<td>#</td>
<td>INQUIRE HOST</td>
</tr>
<tr>
<td>#</td>
<td>SET HOST</td>
</tr>
</tbody>
</table>
Appendix C. EXEC CICS CREATE RESP2 values

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 Table 7 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 8 on page 688, Table 9 on page 689, Table 10 on page 690, Table 11 on page 692, Table 12 on page 693, and Table 13 on page 694.

Table 7. RESP2 values corresponding to messages

<table>
<thead>
<tr>
<th>RESP2</th>
<th>Msgid</th>
<th>Description or message</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n ),400</td>
<td>DFHCA5211</td>
<td>A misplaced delimiter occurs in ATTRIBUTES. The invalid delimiter is at offset ( n ) in the ATTRIBUTES string.</td>
</tr>
<tr>
<td>( n ),401</td>
<td>DFHCA5204</td>
<td>A keyword specified within ATTRIBUTES is invalid. The invalid keyword is at offset ( n ) in the ATTRIBUTES string.</td>
</tr>
<tr>
<td>( n ),402</td>
<td>DFHCA5212, DFHCA5213</td>
<td>A keyword within ATTRIBUTES cannot be uniquely identified from its abbreviation. The invalid keyword is at offset ( n ) in the ATTRIBUTES string.</td>
</tr>
<tr>
<td>( k ),403</td>
<td>DFHCA5501</td>
<td>A required keyword is omitted. The omitted keyword has code ( k ) in Table 8, Table 9, Table 10, Table 11, Table 12, or Table 13.</td>
</tr>
<tr>
<td>( k ),404</td>
<td>DFHCA5529</td>
<td>A required keyword is omitted. The omitted keyword must be selected from two mutually exclusive keywords, as specified in the associated message.</td>
</tr>
<tr>
<td>( k ),405</td>
<td>DFHCA5504</td>
<td>One specified keyword requires another one to be specified. The omitted keyword has code ( k ) in Table 8, Table 9, Table 10, Table 11, Table 12, or Table 13.</td>
</tr>
<tr>
<td>( k ),406</td>
<td>DFHCA5506</td>
<td>A keyword occurs more than once within ATTRIBUTES. The duplicate keyword has code ( k ) in Table 8, Table 9, Table 10, Table 11, Table 12, or Table 13.</td>
</tr>
<tr>
<td>( k ),407</td>
<td>DFHCA5503, DFHCA5506</td>
<td>Conflicting keywords are specified. The keyword causing the conflict has code ( k ) in Table 8, Table 9, Table 10, Table 11, Table 12, or Table 13.</td>
</tr>
<tr>
<td>( k ),410</td>
<td>DFHCA5510, DFHCA5519, DFHCA5521, DFHCA5522, DFHCA5526, DFHCA5528, DFHCA5532, DFHCA5547</td>
<td>An invalid operand is supplied for a keyword within ATTRIBUTES. The keyword in error has code ( k ) in Table 8, Table 9, Table 10, Table 11, Table 12, or Table 13.</td>
</tr>
<tr>
<td>( k ),410</td>
<td>DFHCA5542</td>
<td>Length of Remoteprefix and length of Prefix must be the same. The keyword in error has code ( k ) in Table 8, Table 9, Table 10, Table 11, Table 12, or Table 13.</td>
</tr>
<tr>
<td>( k ),410</td>
<td>DFHCA5543</td>
<td>Generics must be in the same place in the Prefix and in the Remoteprefix. The keyword in error has code ( k ) in Table 8, Table 9, Table 10, Table 11, Table 12, or Table 13.</td>
</tr>
<tr>
<td>( k ),411</td>
<td>DFHCA5507</td>
<td>An operand is supplied for a keyword that does not need one. The keyword in error has code ( k ) in Table 8, Table 9, Table 10, Table 11, Table 12, or Table 13.</td>
</tr>
</tbody>
</table>
Table 7. RESP2 values corresponding to messages (continued)

<table>
<thead>
<tr>
<th>RESP2</th>
<th>Msgid</th>
<th>Description or message</th>
</tr>
</thead>
<tbody>
<tr>
<td>k,412</td>
<td>DFHCA5205</td>
<td>A required operand for a keyword within ATTRIBUTES is omitted. The keyword in error has code k in Table 8, Table 9, Table 10, Table 11, or Table 12.</td>
</tr>
<tr>
<td>k,412</td>
<td>DFHCA5544</td>
<td>The value must be specified as generic because a previous value is generic. The keyword in error has code k in Table 8, Table 9, Table 10, Table 11, Table 12, or Table 13.</td>
</tr>
<tr>
<td>k,413</td>
<td>DFHCA5517</td>
<td>The operands of two or more keywords conflict with one another. The first conflicting keyword detected has code k in Table 8, Table 9, Table 10, Table 11, Table 12, or Table 13.</td>
</tr>
<tr>
<td>k,414</td>
<td>DFHCA5507</td>
<td>The value of the operand of a keyword within ATTRIBUTES is too small. The keyword in error has code k in Table 8, Table 9, Table 10, Table 11, Table 12, or Table 13.</td>
</tr>
<tr>
<td>k,415</td>
<td>DFHCA5513</td>
<td>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 k in Table 8, Table 9, Table 10, Table 11, Table 12, or Table 13.</td>
</tr>
<tr>
<td>k,416</td>
<td>DFHCA5509</td>
<td>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 k in Table 8, Table 9, Table 10, Table 11, Table 12, or Table 13.</td>
</tr>
<tr>
<td>417</td>
<td>DFHCA5523</td>
<td>The specified resource cannot be created with this command.</td>
</tr>
<tr>
<td></td>
<td>DFHCA5535</td>
<td>The resource name is reserved for CICS use.</td>
</tr>
<tr>
<td>418</td>
<td>DFHCA5527</td>
<td>CICS internal programs (whose names begin with DFH) cannot be given attributes that specify remote execution.</td>
</tr>
<tr>
<td>k,419</td>
<td>DFHCA5217</td>
<td>A closing parenthesis has been omitted from a DESCRIPTION keyword within ATTRIBUTES. The keyword in error (DESCRIPTION) has code k in Table 8, Table 9, Table 10, Table 11, Table 12, or Table 13.</td>
</tr>
<tr>
<td>420</td>
<td>DFHCA5508</td>
<td>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.</td>
</tr>
<tr>
<td>421</td>
<td>DFHCA5544</td>
<td>Value must be specified as generic because a previous value is generic.</td>
</tr>
<tr>
<td>422</td>
<td>DFHCA5546</td>
<td>Incorrect TYPE specified for REQUESTMODEL.</td>
</tr>
<tr>
<td>423</td>
<td>DFHCA5548</td>
<td>The option is invalid for a back level request model.</td>
</tr>
<tr>
<td>424</td>
<td>DFHCA5549</td>
<td>The values specified for the two attributes must not be the same.</td>
</tr>
<tr>
<td>425</td>
<td>DFHCA5551</td>
<td>keyword1 cannot be specified as generic unless keyword2 is also generic. See the associated message to determine which keywords are in error.</td>
</tr>
<tr>
<td>427</td>
<td>DFHCA5553</td>
<td>The specified attribute field cannot contain a character as shown.</td>
</tr>
</tbody>
</table>

**Codes caused by errors deleting existing resources**

| 500 | DFHAM4803 | Install failed because the resource is currently in use. |
|     | DFHAM4834 | Install failed because not disabled. |
|     | DFHAM4836 | Another DB2TRAN is installed with the same transid, or failed to install implicit DB2TRAN because there is an existing DB2TRAN installed with the same transid. |
|     | DFHAM4874 | Install failed because the attribute already exists. |
| 500 | DFHAM4894 | Install failed because the resource is not disabled. |
### Table 7. RESP2 values corresponding to messages (continued)

<table>
<thead>
<tr>
<th>RESP2</th>
<th>Msgid</th>
<th>Description or message</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>DFHAM4903</td>
<td>Install failed because the service is open.</td>
</tr>
<tr>
<td>501</td>
<td>DFHAM4841</td>
<td>Install failed because definition of restype resname is in use by task no. taskno (task id. tranid).</td>
</tr>
<tr>
<td>501</td>
<td>DFHZC5980</td>
<td>Resource resource is in use by task taskid Transaction tranid.</td>
</tr>
<tr>
<td>502</td>
<td>DFHZC6304</td>
<td>Deletion of remote terminal termid failed because it is in use by another transaction.</td>
</tr>
<tr>
<td>503</td>
<td>DFHZC5915</td>
<td>Deletion of restype id failed. It needs to be set out of service.</td>
</tr>
<tr>
<td>504</td>
<td>DFHAM4899</td>
<td>Install specified a resource that cannot be replaced.</td>
</tr>
<tr>
<td></td>
<td>DFHZC5998</td>
<td></td>
</tr>
<tr>
<td>505</td>
<td>DFHZC5916</td>
<td>Deletion of terminal termid failed. It has pending DFHZCP activity.</td>
</tr>
<tr>
<td>505</td>
<td>DFHZC5918</td>
<td>Deletion of terminal termid Console consname failed. It has pending DFHZCP activity.</td>
</tr>
<tr>
<td>506</td>
<td>DFHZC5914</td>
<td>Deletion of terminal termid found another deletion of it in progress.</td>
</tr>
<tr>
<td>506</td>
<td>DFHZC5937</td>
<td>Deletion of modename modename found another deletion of it in progress.</td>
</tr>
<tr>
<td>507</td>
<td>DFHZC5902</td>
<td>Deletion of terminal termid failed. BMS Paging session still active.</td>
</tr>
<tr>
<td>508</td>
<td>DFHZC5917</td>
<td>Deletion of terminal termid failed. Error message writer still active.</td>
</tr>
<tr>
<td>509</td>
<td>DFHZC5904</td>
<td>Deletion of terminal termid failed. CEDF is still active.</td>
</tr>
<tr>
<td>510</td>
<td>DFHZC5941</td>
<td>Install for terminal termid failed. Console consname has a conversation outstanding.</td>
</tr>
<tr>
<td>511</td>
<td>DFHZC5907</td>
<td>Deletion of remote shipped terminal failed for connection cccc.</td>
</tr>
<tr>
<td>512</td>
<td>DFHZC5925</td>
<td>Deletion of connection cccc failed. Its AID-Chains are not empty.</td>
</tr>
<tr>
<td>513</td>
<td>DFHZC5929</td>
<td>Deletion of connection cccc failed. It is in use by n indirect connections.</td>
</tr>
<tr>
<td>514</td>
<td>DFHZC5938</td>
<td>Deletion of modename modename failed. Unable to delete sessions.</td>
</tr>
<tr>
<td>515</td>
<td>DFHZC5951</td>
<td>Deletion of connection ssss failed. Unable to delete sessions.</td>
</tr>
<tr>
<td>516</td>
<td>DFHZC5945</td>
<td>Deletion of sessions ssss failed. Connection cccc is defined to IRC.</td>
</tr>
<tr>
<td>517</td>
<td>DFHZC5952</td>
<td>Deletion of terminal termid failed. It needs to be SET RELEASED.</td>
</tr>
<tr>
<td>518</td>
<td>DFHZC5969</td>
<td>Deletion of dependent modenames failed for connection modename.</td>
</tr>
<tr>
<td>519</td>
<td>DFHZC5974</td>
<td>Deletion of pool pppp failed. Unable to delete pool entries.</td>
</tr>
<tr>
<td>520</td>
<td>DFHZC5979</td>
<td>Deletion of pool pppp failed. It still has session termid.</td>
</tr>
<tr>
<td>520</td>
<td>DFHZC5982</td>
<td>Deletion of pool pppp failed. Pool entry is in use for termid.</td>
</tr>
<tr>
<td>521</td>
<td>DFHZC5958</td>
<td>Install failed for xxxx. This is the name of the local system, which must not be replaced.</td>
</tr>
<tr>
<td>522</td>
<td>DFHZC5940</td>
<td>Install for terminal termid failed. Error console cannot be deleted.</td>
</tr>
<tr>
<td>524</td>
<td>DFHZC5943</td>
<td>MRO connection consname cannot be deleted because IRC is open.</td>
</tr>
</tbody>
</table>

#### Codes caused by errors in installing the new resource

<table>
<thead>
<tr>
<th>RESP2</th>
<th>Msgid</th>
<th>Description or message</th>
</tr>
</thead>
<tbody>
<tr>
<td>600</td>
<td>DFHTO6000</td>
<td>The definition for TERMINAL termdef refers to an undefined TYPETERM termtype.</td>
</tr>
<tr>
<td>600</td>
<td>DFHTO6001</td>
<td>The definition for pooled TERMINAL termdef refers to an undefined TYPETERM termtype.</td>
</tr>
</tbody>
</table>

Appendix C. EXEC CICS CREATE RESP2 values
<table>
<thead>
<tr>
<th>RESP2</th>
<th>Msgid</th>
<th>Description or message</th>
</tr>
</thead>
<tbody>
<tr>
<td>601</td>
<td>DFHAM4910</td>
<td>The install failed because the member was not found in the partitioned data set.</td>
</tr>
<tr>
<td>601</td>
<td>DFHTO6002</td>
<td>The definition for SESSIONs sesdef refers to an undefined CONNECTION condef.</td>
</tr>
<tr>
<td>601</td>
<td>DFHZC5911</td>
<td>Install for resource resource failed. Connection cccc not found.</td>
</tr>
<tr>
<td>601</td>
<td>DFHZC5932</td>
<td>Install for modename modename failed. Connection cccc not found.</td>
</tr>
<tr>
<td>602</td>
<td>DFHZC5962</td>
<td>Install for resource resource failed. Modename parameter not found.</td>
</tr>
<tr>
<td>603</td>
<td>DFHZC5906</td>
<td>Install failed because xxxx is not a permitted value for a terminal or connection name.</td>
</tr>
<tr>
<td>604</td>
<td>DFHZC5933</td>
<td>Install for modename modename failed. Connection cccc is not valid here.</td>
</tr>
<tr>
<td>605</td>
<td>DFHAM4889</td>
<td>Install of resource failed because an attribute is invalid.</td>
</tr>
<tr>
<td>606</td>
<td>DFHAM4890</td>
<td>Install of TDQUEUE tdqname failed because the TYPE has not been specified.</td>
</tr>
<tr>
<td>607</td>
<td>DFHAM4870</td>
<td>Install failed for program proname - language RPG is not supported under MVS.</td>
</tr>
<tr>
<td>608</td>
<td>DFHAM4832</td>
<td>Unable to open TDQUEUE tdqname because the DFHINTRA data set is not open.</td>
</tr>
<tr>
<td>608</td>
<td>DFHAM4909</td>
<td>The install failed because the DNAME was not found.</td>
</tr>
<tr>
<td>609</td>
<td>DFHAM4908</td>
<td>The installation failed because the templatename already exists.</td>
</tr>
<tr>
<td>610</td>
<td>DFHAM4905</td>
<td>The installation failed because the option is not available in this system.</td>
</tr>
<tr>
<td>611</td>
<td>DFHAM4901</td>
<td>The installation of a REQUESTMODEL definition failed because a duplicate pattern already exists.</td>
</tr>
<tr>
<td>612</td>
<td>DFHAM4920</td>
<td>The installation of a resource failed because it already exists.</td>
</tr>
<tr>
<td>613</td>
<td>DFHAM4921</td>
<td>The installation of a CORBASESERVER or DJAR definition failed because an invalid JNDIPrefix, Shelf or Certificate value has been specified.</td>
</tr>
<tr>
<td>614</td>
<td>DFHAM4923</td>
<td>The installation of a DJAR definition failed because the specified CorbaServer does not exist.</td>
</tr>
<tr>
<td>615</td>
<td>DFHAM4926</td>
<td>The installation of a CORBASESERVER or DJAR definition failed because the resource resolution transaction CEJR could not attach.</td>
</tr>
<tr>
<td>616</td>
<td>DFHAM4927</td>
<td>The installation of a DJAR definition failed because its HFSFILE is a duplicate of one that already exists.</td>
</tr>
<tr>
<td>617</td>
<td>DFHAM4926</td>
<td>The installation of a DJAR definition failed because the specified CORBASESERVER is not in a valid state.</td>
</tr>
<tr>
<td>618</td>
<td>DFHAM4902</td>
<td>The installation of a CORBASESERVER definition or of a REQUESTMODEL definition failed because it is not a valid definition for this CICS level.</td>
</tr>
<tr>
<td>619</td>
<td>DFHAM4925</td>
<td>The installation of a CORBASESERVER failed because one of its associated TCPIPSERVICES was not installed.</td>
</tr>
<tr>
<td>620</td>
<td>DFHAM4912</td>
<td>The installation of the resource failed because the attribute specified is obsolete.</td>
</tr>
<tr>
<td>620</td>
<td>DFHZC5912</td>
<td>Install for terminal termid failed. It is incompatible with connection cccc.</td>
</tr>
<tr>
<td>620</td>
<td>DFHZC5949</td>
<td>Install for sessions ssss failed. It is incompatible with connection cccc.</td>
</tr>
<tr>
<td>621</td>
<td>DFHZC5900</td>
<td>System sysid has shipped definitions but connection cccc is not known to this system.</td>
</tr>
<tr>
<td>622</td>
<td>DFHZC5921</td>
<td>Install of terminal termid failed. VTAM support not loaded.</td>
</tr>
<tr>
<td>RESP2</td>
<td>Msgid</td>
<td>Description or message</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>622</td>
<td>DFHZC5988</td>
<td>Install for resource resource failed. VTAM support not generated.</td>
</tr>
<tr>
<td>623</td>
<td>DFHZC5909</td>
<td>Install of resource resource failed. Call to DFHIRP <em>irp_function Return_code</em> did not succeed, See DFHIRDS for return code.</td>
</tr>
<tr>
<td>624</td>
<td>DFHZC5931</td>
<td>Install for modename modename failed. Maximum number of APPC sessions would have been exceeded.</td>
</tr>
<tr>
<td></td>
<td>625</td>
<td>DFHAM4929 A resource was not installed because of conflicting attributes.</td>
</tr>
<tr>
<td></td>
<td>625</td>
<td>DFHZC5973 Install for sessions ssss failed. Max session-count reached for modename modename.</td>
</tr>
<tr>
<td></td>
<td>626</td>
<td>DFHAM4930 A URIMAP resource was not installed as it maps the same URI as an existing installed URIMAP.</td>
</tr>
<tr>
<td></td>
<td>626</td>
<td>DFHZC5955 SESNUMB greater than DLTHRED in the SIT (nnnn).</td>
</tr>
<tr>
<td></td>
<td>627</td>
<td>DFHAM4931 The installation of a WEBSERVICE failed because the associated WSBIND file or PIPELINE does not exist.</td>
</tr>
<tr>
<td></td>
<td>627</td>
<td>DFHZC5934 Install for modename modename failed. Single-session connection cccc is already in use.</td>
</tr>
<tr>
<td></td>
<td>628</td>
<td>DFHAM4932 The installation of a WEBSERVICE failed because:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• the associated files in HFS are not correctly set up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• or the associated PIPELINE is not correctly set up</td>
</tr>
<tr>
<td></td>
<td>628</td>
<td>DFHZC5936 Install for modename modename failed. Connection cccc has active modegroup xxxx.</td>
</tr>
<tr>
<td></td>
<td>629</td>
<td>DFHAM4933 The installation of a PIPELINE failed because the file specified in the WSDIR attribute was inaccessible.</td>
</tr>
<tr>
<td></td>
<td>629</td>
<td>DFHZC5939 Install for name failed. Duplicate session- or modegroup-name for connection sysid.</td>
</tr>
<tr>
<td></td>
<td>630</td>
<td>DFHZC5946 Install for sessions ssss failed. Connection cccc is defined to IRC.</td>
</tr>
<tr>
<td></td>
<td>630</td>
<td>DFHAM4934 Urimap install failed due to the specified HOSTCODEPAGE and CHARACTERSET combination being invalid.</td>
</tr>
<tr>
<td></td>
<td>630</td>
<td>DFHZC5948 Install for sessions ssss failed. Connection cccc is not suitable for IRC.</td>
</tr>
<tr>
<td></td>
<td>631</td>
<td>DFHAM4918 Install failed because all cipher codes were rejected.</td>
</tr>
<tr>
<td></td>
<td>632</td>
<td>DFHZC5954 Install for resource resource failed. Unable to install sessions component.</td>
</tr>
<tr>
<td></td>
<td>633</td>
<td>DFHZC5963 <em>operation</em> RUSIZE xxxx from terminal termid was greater than TYPETERM RUSIZE yyyy.</td>
</tr>
<tr>
<td></td>
<td>634</td>
<td>DFHZC5967 Install for modename modename failed. Unable to install sessions.</td>
</tr>
<tr>
<td></td>
<td>635</td>
<td>DFHZC5968 Unable to install LU Services Manager for modename modename.</td>
</tr>
<tr>
<td></td>
<td>636</td>
<td>DFHZC5981 Pool pppp not found.</td>
</tr>
<tr>
<td></td>
<td>637</td>
<td>DFHZC5985 Install for resource resource failed. Unable to install connection component.</td>
</tr>
<tr>
<td></td>
<td>638</td>
<td>DFHTO6003 TERMINAL termdef specifies CONNAME but refers to TYPETERM termtype which does not specify DEVICE=CONSOLE.</td>
</tr>
<tr>
<td></td>
<td>639</td>
<td>DFHTO6004 TERMINAL termdef does not specify CONNAME but refers to TYPETERM termtype which specifies DEVICE=CONSOLE.</td>
</tr>
<tr>
<td></td>
<td>640</td>
<td>DFHTO6005 PRINTER or ALTPRINTER for TERMINAL termdef is invalid for the DEVICE specified in TYPETERM termtype.</td>
</tr>
<tr>
<td></td>
<td>641</td>
<td>DFHTO6006 PRINTERCOPY or ALTPRINTERCOPY for TERMINAL termdef is invalid for the DEVICE specified in TYPETERM termtype.</td>
</tr>
<tr>
<td></td>
<td>642</td>
<td>DFHTO6007 AUTINSTMODEL YESONLY for TERMINAL termdef is invalid for the DEVICE specified in TYPETERM termtype.</td>
</tr>
<tr>
<td></td>
<td>643</td>
<td>DFHTO6008</td>
</tr>
<tr>
<td>RESP2</td>
<td>Msgid</td>
<td>Description or message</td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>644</td>
<td>DFHTO6009</td>
<td>The definition for SESSIONs sesdef refers to CONNECTION condef which specifies a different PROTOCOL.</td>
</tr>
<tr>
<td>645</td>
<td>DFHTO6010</td>
<td>The definition for SESSIONs sesdef must specify PROTOCOL LU6.1 as it refers to an MRO CONNECTION condef.</td>
</tr>
<tr>
<td>646</td>
<td>DFHTO6011</td>
<td>SESSIONs sesdef must specify both SENDCOUNT and RECEIVECOUNT as it refers to an MRO CONNECTION condef.</td>
</tr>
<tr>
<td>647</td>
<td>DFHTO6013</td>
<td>No SESSIONs definition refers to CONNECTION condef.</td>
</tr>
<tr>
<td>648</td>
<td>DFHTO6014</td>
<td>POOL is required for TERMINAL termdef as it refers to TYPETERM typedef which specifies SESSIONTYPE=PIPELINE.</td>
</tr>
<tr>
<td>649</td>
<td>DFHTO6015</td>
<td>TRANSACTION for TERMINAL termdef is invalid for the DEVICE specified in TYPETERM typedef.</td>
</tr>
<tr>
<td>650</td>
<td>DFHTO6016</td>
<td>The MRO CONNECTION condef is referenced by more than one SESSIONs definition, including sesdef.</td>
</tr>
<tr>
<td>651</td>
<td>DFHTO6017</td>
<td>REMOTESYSTEM for TERMINAL termid is invalid for the DEVICE specified in TYPETERM typeterm.</td>
</tr>
<tr>
<td>652</td>
<td>DFHTO6018</td>
<td>TERMINAL termid refers to TYPETERM typeterm which has an invalid ALTSCREEN.</td>
</tr>
<tr>
<td>653</td>
<td>DFHTO6020</td>
<td>SESSIONS sesdef refers to single-session CONNECTION condef but has an invalid MAXIMUM option specified.</td>
</tr>
<tr>
<td>654</td>
<td>DFHTO6023</td>
<td>Connection definition @.BCH detected. Batch-shared database connections are not supported.</td>
</tr>
<tr>
<td>655</td>
<td>DFHTO6025</td>
<td>The definition for LU6.1 SESSIONs sesdef specifies a send or receive count with no prefix.</td>
</tr>
<tr>
<td>656</td>
<td>DFHZC6301</td>
<td>Install for tttt failed. Duplicate netname netname for resource rrrr found.</td>
</tr>
<tr>
<td>657</td>
<td>DFHZC6302</td>
<td>Install for connection cccc failed. Duplicate netname netname for resource rrrr found.</td>
</tr>
<tr>
<td>658</td>
<td>DFHZC6303</td>
<td>Install for tttt failed. Duplicate netname netname found.</td>
</tr>
<tr>
<td>659</td>
<td>DFHZC6334</td>
<td>Install for tttt failed. A session with the same name already exists.</td>
</tr>
<tr>
<td>660</td>
<td>DFHZC6331</td>
<td>Install for connection tttt failed. Non-VTAM terminal with same name already exists.</td>
</tr>
<tr>
<td>661</td>
<td>DFHZC6332</td>
<td>Install for terminal tttt failed. Non-VTAM terminal with same name already exists.</td>
</tr>
<tr>
<td>662</td>
<td>DFHZC6333</td>
<td>Install for terminal termid failed. Console consname already exists.</td>
</tr>
<tr>
<td>663</td>
<td>DFHZC6330</td>
<td>Install for tttt failed. LDCLIST parameter ldclist not found.</td>
</tr>
<tr>
<td>664</td>
<td>DFHZC6333</td>
<td>INSTALL for modename modename failed. Zero sessions specified.</td>
</tr>
<tr>
<td>665</td>
<td>DFHAM4833</td>
<td>Resource cannot be installed with specified userID because of a security error.</td>
</tr>
<tr>
<td></td>
<td>DFHZC6361</td>
<td>Install for terminal portname with userID userid failed because the preset userID has been revoked.</td>
</tr>
<tr>
<td>667</td>
<td>DFHZC6362</td>
<td>Install for terminal portname with userID userid failed because the preset userID’s group access has been revoked.</td>
</tr>
<tr>
<td>668</td>
<td>DFHZC6363</td>
<td>Install for terminal portname with userID userid failed because the ESM returned an unrecognized response.</td>
</tr>
<tr>
<td>669</td>
<td>DFHZC6364</td>
<td>Install for terminal portname with userID userid failed because the external security manager is inactive.</td>
</tr>
<tr>
<td>670</td>
<td>DFHZC6365</td>
<td>Install for terminal portname with userID userid failed because the userID is not authorized to access this CICS system.</td>
</tr>
<tr>
<td>671</td>
<td>DFHZC6366</td>
<td>Install for terminal portname with userID userid failed because the SECLABEL check failed.</td>
</tr>
<tr>
<td>672</td>
<td>DFHZC6367</td>
<td>Install for terminal termid with userID userid failed because the SECLABEL check failed.</td>
</tr>
</tbody>
</table>
Table 7. RESP2 values corresponding to messages (continued)

<table>
<thead>
<tr>
<th>RESP2</th>
<th>Msgid</th>
<th>Description or message</th>
</tr>
</thead>
<tbody>
<tr>
<td>673</td>
<td>DFHZC6368</td>
<td>Install for terminal <code>portname</code> with <code>userid</code> failed because the external security manager is quiesced.</td>
</tr>
<tr>
<td>674</td>
<td>DFHZC6369</td>
<td>Install for terminal <code>portname</code> failed because national language <code>langcode</code> is invalid.</td>
</tr>
<tr>
<td>675</td>
<td>DFHZC6370</td>
<td>Install for terminal <code>portname</code> failed because national language <code>langcode</code> is unavailable.</td>
</tr>
<tr>
<td>676</td>
<td>DFHZC6371</td>
<td>Install for terminal <code>portname</code> with <code>userid</code> failed because the <code>userid</code> is not authorized to use this <code>portname</code>.</td>
</tr>
<tr>
<td>677</td>
<td>DFHZC5944</td>
<td>Install for <code>type(id)</code> has failed. It would make a loop of connection definitions.</td>
</tr>
<tr>
<td>679</td>
<td>DFHAM4837</td>
<td>Install of <code>DB2ENTRY</code> or <code>DB2TRAN</code> failed because <code>DB2CONN</code> not installed.</td>
</tr>
<tr>
<td>680</td>
<td>DFHAM4850</td>
<td><code>DB2TRAN</code> not installed because refers to a <code>DB2ENTRY</code> that is not installed.</td>
</tr>
<tr>
<td>681</td>
<td>DFHAM4851</td>
<td><code>DB2CONN</code> not installed because of a security error, or <code>DB2ENTRY</code> not installed because of a security error, or <code>DB2TRAN</code> not installed because of a security error.</td>
</tr>
<tr>
<td>682</td>
<td>DFHAM4895</td>
<td>The TST has not been assembled with the migrate option when defining CREATE TSMODEL.</td>
</tr>
</tbody>
</table>

**Codes caused by CICS internal logic errors**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>900</td>
<td><code>DFHTO6012</code> The catalog dataset is not available. RDO function is restricted.</td>
</tr>
<tr>
<td>901</td>
<td><code>DFHAM4872</code> Unable to connect to CICS catalog.</td>
</tr>
<tr>
<td>902</td>
<td><code>DFHAM4873</code> Unable to disconnect the CICS catalog.</td>
</tr>
<tr>
<td>903</td>
<td><code>DFHZC6209</code> Invalid ZC catalog request code <code>xxxx</code>.</td>
</tr>
<tr>
<td>904</td>
<td><code>DFHZC6212</code> Level mismatch with catalog record. <code>DFHBS xxx</code>.</td>
</tr>
<tr>
<td>905</td>
<td><code>DFHAM4898</code> Install failed because sufficient storage could not be obtained. <code>DFHZC5901</code>.</td>
</tr>
<tr>
<td>906</td>
<td><code>DFHZC6200</code> Could not obtain DWE storage.</td>
</tr>
<tr>
<td>907</td>
<td><code>DFHZC6203</code> Unable to obtain DWE action-list storage.</td>
</tr>
<tr>
<td>908</td>
<td><code>DFHZC6214</code> Unable to obtain recovery record storage.</td>
</tr>
<tr>
<td>950</td>
<td><code>DFHZC6202</code> Pattern <code>pattern</code> not valid for builder.</td>
</tr>
<tr>
<td>951</td>
<td><code>DFHZC6204</code> Illegal subpattern definition <code>pattern</code>.</td>
</tr>
<tr>
<td>952</td>
<td><code>DFHZC6205</code> Illegal subpattern definition <code>pattern</code>.</td>
</tr>
<tr>
<td>953</td>
<td><code>DFHZC6206</code> Pattern <code>pattern</code> not valid for destroy.</td>
</tr>
<tr>
<td>954</td>
<td><code>DFHZC6207</code> Catalog key too long or zero. Pattern <code>pattern</code>.</td>
</tr>
<tr>
<td>955</td>
<td><code>DFHZC6213</code> Recovery record abandoned. Key is <code>key</code>.</td>
</tr>
<tr>
<td>956</td>
<td><code>DFHZC6341</code> Loop or ABEND has been detected in <code>inmodule</code> by module <code>bymodule</code>.</td>
</tr>
</tbody>
</table>
### Table 8. Keywords associated with keyword numbers. CREATE CONNECTION through CREATE DB2TRAN

<table>
<thead>
<tr>
<th>No.</th>
<th>CONNECTION</th>
<th>CORBASERVER</th>
<th>DB2CONN</th>
<th>DB2ENTRY</th>
<th>DB2TRAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CONNECTION</td>
<td>CORBASERVER</td>
<td>DB2CONN</td>
<td>DB2ENTRY</td>
<td>DB2TRAN</td>
</tr>
<tr>
<td>5</td>
<td>NETNAME</td>
<td>JNDIPREFIX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>INDSYS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>SECURITYNAME</td>
<td>SHELF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(BINDPASSWORD)*</td>
<td>HOST</td>
<td>MSGQUEUE1</td>
<td>TRANSID</td>
<td>TRANSID</td>
</tr>
<tr>
<td>9</td>
<td>(PORT)*</td>
<td>MSGQUEUE2</td>
<td></td>
<td>ENTRY</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>REMOTESYSTEM</td>
<td>SESSBEANTIME(DAYS)</td>
<td>MSGQUEUE3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>REMOTENAME</td>
<td>SESSBEANTIME(HOURS)</td>
<td>PURGECYCLE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>DESCRIPTION</td>
<td>SESSBEANTIME(MINS)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>QUEUELIMIT</td>
<td>CERTIFICATE</td>
<td>STATSQUEUE</td>
<td>PROTECTNUM</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>MAXQTIME</td>
<td>(SSLPORT)*</td>
<td>TCLIMIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>UNAUTH</td>
<td>THREADLIMIT</td>
<td>THREADLIMIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>AUTHID</td>
<td>AUTHID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>CLIENTCERT</td>
<td>PLAN</td>
<td>PLAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>COMTHREADLIMIT</td>
<td>PLANEXITNAME</td>
<td>PLANEXITNAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>SSLUNAUTH</td>
<td>COMAUTHID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>DJARDIR</td>
<td>SIGNID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>CIPHERS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>REMOTESYSTEM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>(SSL)*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>INSERVICE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>98</td>
<td>AUTOCONNECT</td>
<td>AUTOPUBLISH</td>
<td>CONNECTERROR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>99</td>
<td>PROTOCOL</td>
<td>NONTERMREL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>ACCESSMETHOD</td>
<td>STANDBYMODE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>SINGLESESS</td>
<td>THREADERROR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>DATASTREAM</td>
<td>ACCOUNTREC</td>
<td>ACCOUNTREC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>RECORDFORMAT</td>
<td>AUTHTYPE</td>
<td>AUTHTYPE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>104</td>
<td>ATTACHSEC</td>
<td>DROLLBACK</td>
<td>DROLLBACK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>BINDSECURITY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>106</td>
<td>CONNTYPE</td>
<td>PRIORITY</td>
<td>PRIORITY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>PSRECOVERY</td>
<td>THREADWAIT</td>
<td>THREADWAIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>COMAUTHTYPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>USEDFLTUSER</td>
<td>RESYNCMEMBER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>XLNACTION</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:**

1. Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
2. Keywords in parentheses, such as (BINDPASSWORD), are obsolete but are retained for cross-release compatibility.
<table>
<thead>
<tr>
<th>No.</th>
<th>DJAR</th>
<th>DOCTEMPLATE</th>
<th>ENQMODEL</th>
<th>FILE</th>
<th>JOURNALMODEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DJAR</td>
<td>DOCTEMPLATE</td>
<td>ENQMODEL</td>
<td>FILE</td>
<td>JOURNALMODEL</td>
</tr>
<tr>
<td>5</td>
<td>CORBASERVER</td>
<td>(RESSECNUM)*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>DESCRIPTION</td>
<td>DESCRIPTION</td>
<td>DSNAME</td>
<td>DESCRIPTION</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>TEMPLATE</td>
<td>ENQSCOPE</td>
<td>RECORDSIZE</td>
<td>JOURNALNAME</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>FILE</td>
<td>ENQNAME</td>
<td>KEYLENGTH</td>
<td>STREAMNAME</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>TSQUEUE</td>
<td>JOURNAL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>TDQUEUE</td>
<td>REMOTESYSTEM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>PROGRAM</td>
<td>REMOTENAME</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>EXITPGM</td>
<td>PASSWORD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>DDNAME</td>
<td>LSRPOOLID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>MEMBERNAME</td>
<td>STRINGS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>DATABASES</td>
<td>DATABASES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>INDEXBUFFERS</td>
<td>INDEXBUFFERS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>HFSFILE</td>
<td>FWDRECOVLOG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>DESCRIPTION</td>
<td>NSRGROUP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>MAXNUMRECS</td>
<td>CEDTPOOL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>TABLENAME</td>
<td>TABLENAME</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>STATUS</td>
<td>STATUS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>98</td>
<td>RECOVERY</td>
<td>TYPE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>99</td>
<td>APPENDCRLF</td>
<td>STATUS</td>
<td>OPENTIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>TYPE</td>
<td>DISPOSITION</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>ADD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>BROWSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>DELETE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>104</td>
<td>READ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>UPDATE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>106</td>
<td>JNLSYNCREAD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>JNLSYNCWRITE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>JNLREAD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>109</td>
<td>JNUPDATE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>JNADD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>DSNSHARING</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>RECORDFORMAT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>113</td>
<td>TABLE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>114</td>
<td>BACKUPTYPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>115</td>
<td>RLSACCESS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>116</td>
<td>READINTEG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>117</td>
<td>LOAD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>118</td>
<td>UPDATEMODEL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:**

1. Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.

* Keywords in parentheses, such as (BINDPASSWORD), are obsolete but are retained for cross-release compatibility.
Table 10. Keywords associated with keyword numbers. CREATE LSRPOOL through CREATE PIPELINE.

<table>
<thead>
<tr>
<th>No.</th>
<th>EXEC CICS CREATE command names</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>LSRPOOL</strong></td>
</tr>
<tr>
<td>1</td>
<td>LSRPOOL</td>
</tr>
<tr>
<td>2</td>
<td>(RSL)</td>
</tr>
<tr>
<td>3</td>
<td>MAXKEYLENGTH</td>
</tr>
<tr>
<td>4</td>
<td>SHARELIMITENETWORK</td>
</tr>
<tr>
<td>5</td>
<td>STRINGS</td>
</tr>
<tr>
<td>6</td>
<td>DATA512</td>
</tr>
<tr>
<td>7</td>
<td>DATA1K</td>
</tr>
<tr>
<td>8</td>
<td>DATA2K</td>
</tr>
<tr>
<td>9</td>
<td>DATA4K</td>
</tr>
<tr>
<td>10</td>
<td>DATA8K</td>
</tr>
<tr>
<td>11</td>
<td>DATA12K</td>
</tr>
<tr>
<td>12</td>
<td>DATA16K</td>
</tr>
<tr>
<td>13</td>
<td>DATA20K</td>
</tr>
<tr>
<td>14</td>
<td>DATA24K</td>
</tr>
<tr>
<td>15</td>
<td>DATA28K</td>
</tr>
<tr>
<td>16</td>
<td>DATA32K</td>
</tr>
<tr>
<td>17</td>
<td>LSRPOOLID</td>
</tr>
<tr>
<td>18</td>
<td>DESCRIPTIONI</td>
</tr>
<tr>
<td>19</td>
<td>INDEX512</td>
</tr>
<tr>
<td>20</td>
<td>INDEX1K</td>
</tr>
<tr>
<td>21</td>
<td>INDEX2K</td>
</tr>
<tr>
<td>22</td>
<td>INDEX4K</td>
</tr>
<tr>
<td>23</td>
<td>INDEX8K</td>
</tr>
<tr>
<td>24</td>
<td>INDEX12K</td>
</tr>
<tr>
<td>25</td>
<td>INDEX16K</td>
</tr>
<tr>
<td>26</td>
<td>INDEX20K</td>
</tr>
<tr>
<td>27</td>
<td>INDEX24K</td>
</tr>
<tr>
<td>28</td>
<td>INDEX28K</td>
</tr>
<tr>
<td>29</td>
<td>INDEX32K</td>
</tr>
<tr>
<td>30</td>
<td>HSDATA4K</td>
</tr>
<tr>
<td>31</td>
<td>HSDATA8K</td>
</tr>
<tr>
<td>32</td>
<td>HSDATA12K</td>
</tr>
<tr>
<td>33</td>
<td>HSDATA16K</td>
</tr>
<tr>
<td>34</td>
<td>HSDATA20K</td>
</tr>
<tr>
<td>35</td>
<td>HSDATA24K</td>
</tr>
<tr>
<td>36</td>
<td>HSDATA28K</td>
</tr>
<tr>
<td>37</td>
<td>HSDATA32K</td>
</tr>
<tr>
<td>38</td>
<td>HSINDEX4K</td>
</tr>
<tr>
<td>39</td>
<td>HSINDEX8K</td>
</tr>
<tr>
<td>40</td>
<td>HSINDEX12K</td>
</tr>
<tr>
<td>41</td>
<td>HSINDEX16K</td>
</tr>
<tr>
<td>42</td>
<td>HSINDEX20K</td>
</tr>
<tr>
<td>43</td>
<td>HSINDEX24K</td>
</tr>
<tr>
<td>44</td>
<td>HSINDEX28K</td>
</tr>
<tr>
<td>45</td>
<td>HSINDEX32K</td>
</tr>
<tr>
<td>46</td>
<td>STATUS</td>
</tr>
<tr>
<td>47</td>
<td>RESIDENT</td>
</tr>
<tr>
<td>48</td>
<td>USAGE</td>
</tr>
<tr>
<td>49</td>
<td>USELPACOPY</td>
</tr>
</tbody>
</table>

Note:

1. Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
* Keywords in parentheses, such as (BINDPASSWORD), are obsolete but are retained for cross-release compatibility.
### Table 11. Keywords associated with keyword numbers. CREATE PROCESSTYPE through CREATE SESSIONS.

<table>
<thead>
<tr>
<th>No.</th>
<th>EXEC CICS CREATE command names</th>
<th>KEYWORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PROCESSTYPE PROFILE PROGRAM REQUESTMODEL SESSIONS</td>
<td><strong>PROCESSTYPE</strong> PROFILE PROGRAM REQUESTMODEL SESSIONS</td>
</tr>
<tr>
<td>5</td>
<td>AUDITLOG MODENAME (RSL)*</td>
<td><strong>AUDITLOG</strong> MODENAME (RSL)*</td>
</tr>
<tr>
<td>6</td>
<td>DESCRIPTION JOURNAL DESCRIPTION DESCRIPTION SESSNAME</td>
<td><strong>DESCRIPTION</strong> JOURNAL DESCRIPTION DESCRIPTION SESSNAME</td>
</tr>
<tr>
<td>7</td>
<td>FILE NEPCLASS REMOTESYSTEM</td>
<td><strong>FILE</strong> NEPCLASS REMOTESYSTEM</td>
</tr>
<tr>
<td>8</td>
<td>RTIMOUT REMOTENAME (OMGINTERFACE)* MODENAME</td>
<td><strong>RTIMOUT</strong> REMOTENAME (OMGINTERFACE)* MODENAME</td>
</tr>
<tr>
<td>9</td>
<td>DESCRIPTION TRANSID (OMGOPERATION)* MAXIMUM</td>
<td><strong>DESCRIPTION</strong> TRANSID (OMGOPERATION)* MAXIMUM</td>
</tr>
<tr>
<td>10</td>
<td>FACILITYLIKE JVMCLASS TRANSID</td>
<td><strong>FACILITYLIKE</strong> JVMCLASS TRANSID</td>
</tr>
<tr>
<td>11</td>
<td>CORBASERVER RECEIVEPFX</td>
<td><strong>CORBASERVER</strong> RECEIVEPFX</td>
</tr>
<tr>
<td>12</td>
<td>BEANNAME RECEIVECOUNT</td>
<td><strong>BEANNAME</strong> RECEIVECOUNT</td>
</tr>
<tr>
<td>13</td>
<td>MODULE SENDPFX</td>
<td><strong>MODULE</strong> SENDPFX</td>
</tr>
<tr>
<td>14</td>
<td>INTERFACE SENDCOUNT</td>
<td><strong>INTERFACE</strong> SENDCOUNT</td>
</tr>
<tr>
<td>15</td>
<td>OPERATION (OPERID)*</td>
<td><strong>OPERATION</strong> (OPERID)*</td>
</tr>
<tr>
<td>16</td>
<td>(OPERPRIORITY)*</td>
<td><strong>(OPERPRIORITY)</strong>*</td>
</tr>
<tr>
<td>17</td>
<td>(OPERRSL)*</td>
<td><strong>(OPERRSL)</strong>*</td>
</tr>
<tr>
<td>18</td>
<td>(OPERSECURITY)*</td>
<td><strong>(OPERSECURITY)</strong>*</td>
</tr>
<tr>
<td>19</td>
<td>USERID</td>
<td><strong>USERID</strong></td>
</tr>
<tr>
<td>20</td>
<td>SENDSIZE</td>
<td><strong>SENDSIZE</strong></td>
</tr>
<tr>
<td>21</td>
<td>RECEIVESIZE</td>
<td><strong>RECEIVESIZE</strong></td>
</tr>
<tr>
<td>22</td>
<td>(TRANSACTION)*</td>
<td><strong>(TRANSACTION)</strong>*</td>
</tr>
<tr>
<td>23</td>
<td>SESSPRIORITY</td>
<td><strong>SESSPRIORITY</strong></td>
</tr>
<tr>
<td>24</td>
<td>USERAREALEN</td>
<td><strong>USERAREALEN</strong></td>
</tr>
<tr>
<td>25</td>
<td>IOAREALEN</td>
<td><strong>IOAREALEN</strong></td>
</tr>
<tr>
<td>27</td>
<td>NEPCLASS</td>
<td><strong>NEPCLASS</strong></td>
</tr>
<tr>
<td>28</td>
<td>DESCRIPTION</td>
<td><strong>DESCRIPTION</strong></td>
</tr>
<tr>
<td>33</td>
<td>JVMPROFILE</td>
<td><strong>JVMPROFILE</strong></td>
</tr>
<tr>
<td>97</td>
<td>STATUS SCRNSIZE LANGUAGE INIFACETYPE</td>
<td><strong>STATUS</strong> SCRNSIZE LANGUAGE INIFACETYPE</td>
</tr>
<tr>
<td>99</td>
<td>AUDITLEVEL MSGJRNAL RELOAD</td>
<td><strong>AUDITLEVEL</strong> MSGJRNAL RELOAD</td>
</tr>
<tr>
<td>100</td>
<td>USERRECORDS MSGINTEG RESIDENT</td>
<td><strong>USERRECORDS</strong> MSGINTEG RESIDENT</td>
</tr>
<tr>
<td>101</td>
<td>ONEWTE USAGE</td>
<td><strong>ONEWTE</strong> USAGE</td>
</tr>
<tr>
<td>102</td>
<td>(PROTECT)* USELPCOPY</td>
<td><strong>(PROTECT)</strong>* USELPCOPY</td>
</tr>
<tr>
<td>103</td>
<td>DVSPRRT CEDF</td>
<td><strong>DVSPRRT</strong> CEDF</td>
</tr>
<tr>
<td>104</td>
<td>INBFMH DATALOCATION</td>
<td><strong>INBFMH</strong> DATALOCATION</td>
</tr>
<tr>
<td>105</td>
<td>RAQ EXECKEY</td>
<td><strong>RAQ</strong> EXECKEY</td>
</tr>
<tr>
<td>106</td>
<td>LOGREC</td>
<td><strong>LOGREC</strong></td>
</tr>
<tr>
<td>107</td>
<td>PRINTERCOMP EXECUTIONSET</td>
<td><strong>PRINTERCOMP</strong> EXECUTIONSET</td>
</tr>
<tr>
<td>108</td>
<td>CHAINCONTROL DYNAMIC</td>
<td><strong>CHAINCONTROL</strong> DYNAMIC</td>
</tr>
<tr>
<td>109</td>
<td>UCTRAN CONCURRENCY</td>
<td><strong>UCTRAN</strong> CONCURRENCY</td>
</tr>
<tr>
<td>110</td>
<td>JVM</td>
<td><strong>JVM</strong></td>
</tr>
<tr>
<td>111</td>
<td>(HOTPOOL)</td>
<td><strong>(HOTPOOL)</strong></td>
</tr>
<tr>
<td>112</td>
<td>API</td>
<td><strong>API</strong></td>
</tr>
</tbody>
</table>

**Note:**

1. Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
2. Keywords in parentheses, such as (BINDPASSWORD), are obsolete but are retained for cross-release compatibility.
### Table 12. Keywords associated with keyword numbers. CREATE TCPIPSERVICE through CREATE TRANSACTION.

<table>
<thead>
<tr>
<th>No.</th>
<th>TCPIPSERVICE</th>
<th>TDQUEUE</th>
<th>TERMINAL</th>
<th>TRANCLASS</th>
<th>TRANSACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TCPIPSERVICE</td>
<td>TDQUEUE</td>
<td>TERMINAL</td>
<td>TRANCLASS</td>
<td>TRANSACTION</td>
</tr>
<tr>
<td>5</td>
<td>MAXACTIVE</td>
<td></td>
<td></td>
<td>(RSL)*</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>DESCRIPTION</td>
<td>AUTINSTNAME</td>
<td>DESCRIPTION</td>
<td>PROGRAM</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>BLOCKSIZE</td>
<td>TYPETERM</td>
<td>PURGETHRESH</td>
<td>TWASIZE</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>DATABASE</td>
<td>NETNAME</td>
<td>PROFILE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>DDNAME</td>
<td>(CONSOLE)</td>
<td>PARTITIONSET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>DSNAME</td>
<td>REMOTESYSTEM</td>
<td>REMOTESYSTEM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>RECORDSIZE</td>
<td>REMOTENAME</td>
<td>REMOTENAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>FACILITYID</td>
<td>MODENAME</td>
<td>PRIORITY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>TRANSID</td>
<td>PRINTER</td>
<td>(TCLASS)*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>URM</td>
<td>TRIGGERLEVEL</td>
<td>ALTPRINTER</td>
<td>TASKREQ</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>PORTNUMBER</td>
<td>USERID</td>
<td>(OPERID)*</td>
<td>XTRANID</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>CIPHERS</td>
<td>INDIRECTNAME</td>
<td>(OPERPRIORITY)*</td>
<td>DTIMOUT</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>CERTIFICATE</td>
<td>REMOTENAME</td>
<td>(OPERRSL)*</td>
<td>(TRANSEC)*</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>TRANSACTION</td>
<td>REMOTESYSTEM</td>
<td>(OPERSECURITY)*</td>
<td>TRPROF</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>BACKLOG</td>
<td>SYSOUTCLASS</td>
<td>USERID</td>
<td>(PRIMEDSIZE)*</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>REMOTELENGTH</td>
<td>POOL</td>
<td>ALIAS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>TASKLIMIT</td>
<td></td>
<td>DESCRIPTION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>TSQPREFIX</td>
<td>TRANSACTION</td>
<td>TPNNAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>IPADDRESS</td>
<td>TERMIPRIORITY</td>
<td>XTPNAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>SOCKETCLOSE</td>
<td></td>
<td>TRANCLASS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>DNSGROUP</td>
<td></td>
<td>RUNAWAY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td>SECURITYNAME</td>
<td>WAITTIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>MAXDATALEN</td>
<td>(BINDPASSWORD)*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>DESCRIPTION</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td>NATLANG</td>
<td>BREXIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>CONSNAME</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td>REMOTESYSNET</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>97</td>
<td></td>
<td>INSERVICE</td>
<td>STATUS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>98</td>
<td>TYPE</td>
<td>PRINTERCOPY</td>
<td>LOCALQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>99</td>
<td>DISPOSITION</td>
<td>ALTPRINCOPY</td>
<td>(INDOUBT)*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>ERROROPTION</td>
<td>AUTINSTMODEL</td>
<td>RESTART</td>
<td></td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>OPENTIME</td>
<td></td>
<td>SPURGE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>RECORDFORMAT</td>
<td>ATTACHSEC</td>
<td>TPURGE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>BLOCKFORMAT</td>
<td>BINDSECURITY</td>
<td>DUMP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>104</td>
<td>STATUS</td>
<td>REWIND</td>
<td>USEDFLTUSER</td>
<td>(EXTSEC)*</td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>SSL</td>
<td>TYPEFILE</td>
<td>RESSEC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>106</td>
<td>PROTOCOL</td>
<td>ATIFACILITY</td>
<td>TRACE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>RECOVSTATUS</td>
<td></td>
<td>DYNAMIC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>WAITACTION</td>
<td></td>
<td>CMDSEC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>109</td>
<td>AUTHENTICATE</td>
<td>PRINTCONTROL</td>
<td>TASKDATALOC</td>
<td>TASKDATAKEY</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>GRPCRITICAL</td>
<td>WAIT</td>
<td>STORAGECLEAR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>ATTACHSEC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>PRIVACY</td>
<td></td>
<td>SHUTDOWN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>113</td>
<td></td>
<td>ISOLATE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>114</td>
<td></td>
<td>CONFDATA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>115</td>
<td></td>
<td>WAIT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>116</td>
<td></td>
<td>ACTION</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>117</td>
<td></td>
<td>ROUTABLE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:**

1. Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.
2. * Keywords in parentheses, such as (BINDPASSWORD), are obsolete but are retained for cross-release compatibility.
Table 13. Keywords associated with keyword numbers. CREATE TSMODEL through CREATE WEBSERVICE.

<table>
<thead>
<tr>
<th>No.</th>
<th>EXEC CICS CREATE command names</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TSMODEL</td>
</tr>
<tr>
<td>2</td>
<td>TYPETERM</td>
</tr>
<tr>
<td>3</td>
<td>URIMAP</td>
</tr>
<tr>
<td>4</td>
<td>WEBSERVICE</td>
</tr>
<tr>
<td>5</td>
<td>DEVICE</td>
</tr>
<tr>
<td>6</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>7</td>
<td>TERMMODEL</td>
</tr>
<tr>
<td>8</td>
<td>SESSIONTYPE</td>
</tr>
<tr>
<td>9</td>
<td>POOLNAME</td>
</tr>
<tr>
<td>10</td>
<td>REMOTESYSTEM</td>
</tr>
<tr>
<td>11</td>
<td>LDCLIST</td>
</tr>
<tr>
<td>12</td>
<td>REMOTEPREFIX</td>
</tr>
<tr>
<td>13</td>
<td>DEFSCREEN</td>
</tr>
<tr>
<td>14</td>
<td>XPREFIX</td>
</tr>
<tr>
<td>15</td>
<td>ALTPREFIX</td>
</tr>
<tr>
<td>16</td>
<td>SENDSIZE</td>
</tr>
<tr>
<td>17</td>
<td>RECEIVESIZE</td>
</tr>
<tr>
<td>18</td>
<td>LOGMODE</td>
</tr>
<tr>
<td>19</td>
<td>PAGESIZE</td>
</tr>
<tr>
<td>20</td>
<td>ALTPAGE</td>
</tr>
<tr>
<td>21</td>
<td>ALTSUFFIX</td>
</tr>
<tr>
<td>22</td>
<td>USERAREALEN</td>
</tr>
<tr>
<td>23</td>
<td>IOAREALEN</td>
</tr>
<tr>
<td>24</td>
<td>PROGRAM</td>
</tr>
<tr>
<td>25</td>
<td>PIPELINE</td>
</tr>
<tr>
<td>26</td>
<td>NEPCLASS</td>
</tr>
<tr>
<td>27</td>
<td>WEBSERVICE</td>
</tr>
<tr>
<td>28</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>29</td>
<td>USERID</td>
</tr>
<tr>
<td>30</td>
<td>CIPHERS</td>
</tr>
<tr>
<td>31</td>
<td>CERTIFICATE</td>
</tr>
<tr>
<td>32</td>
<td>MEDIATYPE</td>
</tr>
<tr>
<td>33</td>
<td>CHARACTERSET</td>
</tr>
<tr>
<td>34</td>
<td>HOSTCODEPAGE</td>
</tr>
<tr>
<td>35</td>
<td>TEMPLATENAME</td>
</tr>
<tr>
<td>36</td>
<td>HFSFILE</td>
</tr>
<tr>
<td>37</td>
<td>LOCATION</td>
</tr>
<tr>
<td>38</td>
<td>STATUS</td>
</tr>
<tr>
<td>39</td>
<td>VALIDATION</td>
</tr>
<tr>
<td>40</td>
<td>AUTOCONNECT</td>
</tr>
<tr>
<td>41</td>
<td>SHIPPABLE</td>
</tr>
<tr>
<td>42</td>
<td>APLKBDD</td>
</tr>
<tr>
<td>43</td>
<td>SECURITY</td>
</tr>
<tr>
<td>44</td>
<td>APLTEXT</td>
</tr>
<tr>
<td>45</td>
<td>AUDIBLEALARM</td>
</tr>
<tr>
<td>46</td>
<td>COLOR</td>
</tr>
<tr>
<td>47</td>
<td>COPY</td>
</tr>
<tr>
<td>48</td>
<td>DUALCASEKYBD</td>
</tr>
<tr>
<td>49</td>
<td>EXTENDEDDS</td>
</tr>
<tr>
<td>50</td>
<td>ANALYZER</td>
</tr>
<tr>
<td>51</td>
<td>HILIGHT</td>
</tr>
<tr>
<td>52</td>
<td>KATAKANA</td>
</tr>
<tr>
<td>53</td>
<td>LIGHTPEN</td>
</tr>
<tr>
<td>54</td>
<td>MSCRCONTROL</td>
</tr>
<tr>
<td>55</td>
<td>OBFORMAT</td>
</tr>
<tr>
<td>56</td>
<td>PARTITIONS</td>
</tr>
<tr>
<td>57</td>
<td>PRINTADAPTER</td>
</tr>
<tr>
<td>58</td>
<td>PROGSYMBOLS</td>
</tr>
<tr>
<td>59</td>
<td>VALIDATION</td>
</tr>
<tr>
<td>60</td>
<td>FORMFEED</td>
</tr>
<tr>
<td>61</td>
<td>HORIZFORM</td>
</tr>
<tr>
<td>62</td>
<td>VERTICALFORM</td>
</tr>
<tr>
<td>63</td>
<td>TEXTKYBD</td>
</tr>
<tr>
<td>64</td>
<td>TEXTPRINT</td>
</tr>
<tr>
<td>65</td>
<td>QUERY</td>
</tr>
<tr>
<td>66</td>
<td>OUTLINE</td>
</tr>
<tr>
<td>67</td>
<td>SOSI</td>
</tr>
<tr>
<td>68</td>
<td>BACKTRANS</td>
</tr>
</tbody>
</table>
### Table 13. Keywords associated with keyword numbers (continued). CREATE TSMODEL through CREATE WEBSERVICE.

<table>
<thead>
<tr>
<th>No.</th>
<th>EXEC CICS CREATE command names</th>
</tr>
</thead>
<tbody>
<tr>
<td>125</td>
<td>ASCII</td>
</tr>
<tr>
<td>126</td>
<td>BRACKET</td>
</tr>
<tr>
<td>127</td>
<td>FMHPARM</td>
</tr>
<tr>
<td>128</td>
<td>OBOPERID</td>
</tr>
<tr>
<td>129</td>
<td>AUTOPAGE</td>
</tr>
<tr>
<td>130</td>
<td>ERRLASTLINE</td>
</tr>
<tr>
<td>131</td>
<td>ERRINTENSIFY</td>
</tr>
<tr>
<td>132</td>
<td>ERRCOLOR</td>
</tr>
<tr>
<td>133</td>
<td>ERRHLIGHT</td>
</tr>
<tr>
<td>134</td>
<td>ATI</td>
</tr>
<tr>
<td>135</td>
<td>CREATESESS</td>
</tr>
<tr>
<td>136</td>
<td>RELREQ</td>
</tr>
<tr>
<td>137</td>
<td>DISCREQ</td>
</tr>
<tr>
<td>138</td>
<td>SIGNOFF</td>
</tr>
<tr>
<td>139</td>
<td>ROUTEDMSGS</td>
</tr>
<tr>
<td>140</td>
<td>LOGONMSG</td>
</tr>
<tr>
<td>141</td>
<td>BUILDCHAIN</td>
</tr>
<tr>
<td>142</td>
<td>UCTRAN</td>
</tr>
<tr>
<td>143</td>
<td>TTI</td>
</tr>
<tr>
<td>144</td>
<td>RECOVOPTION</td>
</tr>
<tr>
<td>145</td>
<td>RECOVNOTIFY</td>
</tr>
<tr>
<td>146</td>
<td>(XRFSIGNOFF)*</td>
</tr>
<tr>
<td>147</td>
<td>(LOGEMODECOM)*</td>
</tr>
<tr>
<td>148</td>
<td>RSTSIGNOFF</td>
</tr>
</tbody>
</table>

**Note:**

1. Keyword number 1 always refers to the first operand of the CREATE command; that is, the resource being created.

* Keywords in parentheses, such as (BINDPASSWORD), are obsolete but are retained for cross-release compatibility.
Appendix D. Threadsafe SPI commands

Threadsafe SPI commands
DISCARD DB2CONN
DISCARD DB2ENTRY
DISCARD DB2TRAN
# DISCARD DOCTEMPLATE
INQUIRE DB2CONN
INQUIRE DB2ENTRY
INQUIRE DB2TRAN
# INQUIRE DOCTEMPLATE
INQUIRE EXITPROGRAM
INQUIRE TASK
# INQUIRE WEB
INQUIRE WORKREQUEST
SET DB2CONN
SET DB2ENTRY
SET DB2TRAN
# SET WEB
SET WORKREQUEST

For an introduction to the concept of threadsafe commands see "CICS threadsafe commands in the SPI" on page 29.
The CICS Transaction Server for z/OS library

The published information for CICS Transaction Server for z/OS is delivered in the following forms:

The CICS Transaction Server for z/OS Information Center

The CICS Transaction Server for z/OS Information Center is the primary source of user information for CICS Transaction Server. The Information Center contains:

- Information for CICS Transaction Server in HTML format.
- Licensed and unlicensed CICS Transaction Server books provided as Adobe Portable Document Format (PDF) files. You can use these files to print hardcopy of the books. For more information, see "PDF-only books."
- Information for related products in HTML format and PDF files.

One copy of the CICS Information Center, on a CD-ROM, is provided automatically with the product. Further copies can be ordered, at no additional charge, by specifying the Information Center feature number, 7014.

Licensed documentation is available only to licensees of the product. A version of the Information Center that contains only unlicensed information is available through the publications ordering system, order number SK3T-6945.

Entitlement hardcopy books

The following essential publications, in hardcopy form, are provided automatically with the product. For more information, see "The entitlement set."

The entitlement set

The entitlement set comprises the following hardcopy books, which are provided automatically when you order CICS Transaction Server for z/OS, Version 3 Release 1:

- Memo to Licensees, GI10-2559
- CICS Transaction Server for z/OS Program Directory, GI10-2586
- CICS Transaction Server for z/OS Installation Guide, GC34-6426
- CICS Transaction Server for z/OS Licensed Program Specification, GC34-6608

You can order further copies of the following books in the entitlement set, using the order number quoted above:

- CICS Transaction Server for z/OS Release Guide
- CICS Transaction Server for z/OS Installation Guide
- CICS Transaction Server for z/OS Licensed Program Specification

PDF-only books

The following books are available in the CICS Information Center as Adobe Portable Document Format (PDF) files:

CICS books for CICS Transaction Server for z/OS

General

- CICS Transaction Server for z/OS Program Directory, GI10-2586
- CICS Transaction Server for z/OS Installation Guide
- CICS Transaction Server for z/OS Licensed Program Specification
- CICS Transaction Server for z/OS Migration from CICS TS Version 2.3, GC34-6425
CICS Transaction Server for z/OS Migration from CICS TS Version 1.3
GC34-6423
CICS Transaction Server for z/OS Migration from CICS TS Version 2.2
GC34-6424
CICS Transaction Server for z/OS Installation Guide, GC34-6426

Administration
CICS System Definition Guide, SC34-6428
CICS Customization Guide, SC34-6429
CICS Resource Definition Guide, SC34-6430
CICS Operations and Utilities Guide, SC34-6431
CICS Supplied Transactions, SC34-6432

Programming
CICS Application Programming Guide, SC34-6433
CICS Application Programming Reference, SC34-6434
CICS System Programming Reference, SC34-6435
CICS Front End Programming Interface User's Guide, SC34-6436
CICS C++ OO Class Libraries, SC34-6437
CICS Distributed Transaction Programming Guide, SC34-6438
CICS Business Transaction Services, SC34-6439
Java Applications in CICS, SC34-6440
JCICS Class Reference, SC34-6001

Diagnosis
CICS Problem Determination Guide, SC34-6441
CICS Messages and Codes, GC34-6442
CICS Diagnosis Reference, GC34-6899
CICS Data Areas, GC34-6902
CICS Trace Entries, SC34-6443
CICS Supplementary Data Areas, GC34-6905

Communication
CICS Intercommunication Guide, SC34-6448
CICS External Interfaces Guide, SC34-6449
CICS Internet Guide, SC34-6450

Special topics
CICS Recovery and Restart Guide, SC34-6451
CICS Performance Guide, SC34-6452
CICS IMS Database Control Guide, SC34-6453
CICS RACF Security Guide, SC34-6454
CICS Shared Data Tables Guide, SC34-6455
CICS DB2 Guide, SC34-6457
CICS Debugging Tools Interfaces Reference, GC34-6908

CICSPlex SM books for CICS Transaction Server for z/OS

General
CICSPlex SM Concepts and Planning, SC34-6459
CICSPlex SM User Interface Guide, SC34-6460
CICSPlex SM Web User Interface Guide, SC34-6461

Administration and Management
CICSPlex SM Administration, SC34-6462
CICSPlex SM Operations Views Reference, SC34-6463
CICSPlex SM Monitor Views Reference, SC34-6464
CICSPlex SM Managing Workloads, SC34-6465
CICSPlex SM Managing Resource Usage, SC34-6466
CICSPlex SM Managing Business Applications, SC34-6467

Programming
CICSPlex SM Application Programming Guide, SC34-6468
CICSPlex SM Application Programming Reference, SC34-6469
Diagnosis

CICSPlex SM Resource Tables Reference, SC34-6470
CICSPlex SM Messages and Codes, GC34-6471
CICSPlex SM Problem Determination, GC34-6472

CICS family books

Communication

CICS Family: Interproduct Communication, SC34-6473
CICS Family: Communicating from CICS on System/390, SC34-6474

Licensed publications

The following licensed publications are not included in the unlicensed version of the Information Center:

CICS Diagnosis Reference, GC34-6899
CICS Data Areas, GC34-6902
CICS Supplementary Data Areas, GC34-6905
CICS Debugging Tools Interfaces Reference, GC34-6908

Other CICS books

The following publications contain further information about CICS, but are not provided as part of CICS Transaction Server for z/OS, Version 3 Release 1.

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 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

Books from related libraries

You may also need to refer to relevant MVS publications.

Other publications

- VTAM Programming, SC23-0115
- IMS Database Administration Guide, SC26-4281
- IMS Messages and Codes manual, SC26-4290
- IMS Utilities Reference manual, SC26-4284
- IMS System Administration Guide, SC26-4282
- An Introduction to the IBM 3270 Information Display System, GA27-2739
- MVSDFP V3R3 System Programming Reference, SC26-4567
- z/OS V1R4.0 Security Server RACF Security Administrator’s Guide, SA22-7683
- z/OS DFSMSdfp Advanced Services, SC26-7400
- z/OS V1R1.0 MVS Diagnosis: Procedures, GA22-7587
Determining if a publication is current

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

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

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

Updates to the softcopy are clearly marked by revision codes (usually a # character) to the left of the changes.
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 307
ACCESSMETHOD option
  INQUIRE CONNECTION command 187
  INQUIRE DSNAME command 223
  INQUIRE FILE command 243
  INQUIRE TERMINAL command 361
ACOSTATUS option
  INQUIRE CONNECTION command 187
  INQUIRE TERMINAL command 362
  SET CONNECTION command 481
  SET MODENAME command 552
  SET TERMINAL command 594
ACQUIRE TERMINAL command 35
  conditions 36
ACTION option
  SET DSNAME command 517
  SET SYSDUMPCODE command 579
  SET TRANSDUMPCODE command 613
  SET UOWLINK command 625
ACTIVE option
  INQUIRE MODENAME command 269
  INQUIRE TRANCLASS command 386
ACTIVITY option
  INQUIRE TASK command 330
ACTIVITYID option
  INQUIRE TASK command 330
ACTJVMTCBS option
  INQUIRE DISPATCHER command 214
ACTOPENTCBS option
  INQUIRE DISPATCHER command 214
  INQUIRE SYSTEM command 319
ACTSOCKETS option
  INQUIRE TCPIP command 342
ACTXPTCBS option
  INQUIRE DISPATCHER command 215
ADD option
  INQUIRE FILE command 244
  SET FILE command 532
ADDRESS option
  INQUIRE STORAGE command 308
AFFINITY option
  SET CONNECTION command 481
AFTER option
  INQUIRE REQID command 297
AGE option
  INQUIRE JVM command 263
  INQUIRE UOW command 408
AIBRIDGE option
  INQUIRE AUTOINSTALL command 171
  SET AUTOINSTALL command 474
AKP option
  INQUIRE SYSTEM command 319
  SET SYSTEM command 575
ALIGNED attribute
  PL/I 13
ALL option
  PERFORM STATISTICS command 467
ALLOCAGE option
  INQUIRE JVM command 263
ALTPAGEHT option
  INQUIRE TERMINAL command 362
ALTPAGEWD option
  INQUIRE TERMINAL command 362
ALTPRINTER option
  INQUIRE TERMINAL command 362
  SET TERMINAL command 594
ALTPTCOPYST option
  INQUIRE TERMINAL command 362
  SET TERMINAL command 594
ALTSCRNHT option
  INQUIRE TERMINAL command 362
  SET TERMINAL command 594
APL option
  INQUIRE EXITPROGRAM command 237
  INQUIRE PROGRAM command 288
APLKEYBDST option
  INQUIRE TERMINAL command 362
APLTEXTST option
  INQUIRE TERMINAL command 362
APPENDCRLF option
  INQUIRE DOCTEMPLATE command 219
APPLNAMEST option
  INQUIRE MONITOR command 273
argument lengths 14
argument values
  assembler language 13
  C/370 12
  COBOL 11
  PL/I 12
ASCII option
  INQUIRE TERMINAL command 363
  assembler language
  argument values 13
ASSERTED option
  INQUIRE CORBASERVER command 194
AT option
  INQUIRE REQID command 297
ATIFACILITY option
  INQUIRE TDQUEUE command 351
  SET TDQUEUE command 589
ATISTATUS option
  INQUIRE TERMINAL command 363
  SET TERMINAL command 594
ATITERMID option
  INQUIRE TDQUEUE command 351
  SET TDQUEUE command 589
ATITRANID option
  INQUIRE TDQUEUE command 351
  SET TDQUEUE command 590

ATIUSERID option
  INQUIRE TDQUEUE command 352
  SET TDQUEUE command 590

ATTACHSEC option
  INQUIRE TCPIPSERVICE command 344

ATTACHTIME option
  INQUIRE TASK command 330

ATTRIBUTES option
  CREATE CONNECTION command 49
  CREATE CORBASERVER command 52
  CREATE DB2CONN command 56
  CREATE DB2ENTRY command 59
  CREATE DB2TRAN command 61
  CREATE DJAR command 63
  CREATE DOCTEMPLATE command 65
  CREATE ENQMODEL command 68
  CREATE FILE command 70
  CREATE JOURNALMODEL command 72
  CREATE LSRPOOL command 75
  CREATE MAPSET command 77
  CREATE PARTITIONSET command 79
  CREATE PARTNER command 81
  CREATE PIPELINE command 83
  CREATE PROCESSTYPE command 85
  CREATE PROFILE command 88
  CREATE PROGRAM command 90
  CREATE REQUESTMODEL command 93
  CREATE SESSIONS command 95
  CREATE TCPIPSERVICE command 98
  CREATE TERMINAL command 105
  CREATE TRANCLASS command 107
  CREATE TRANSACTION command 110
  CREATE TSMODEL command 113
  CREATE TYPETERM command 116
  CREATE UIMAP command 119
  CREATE WEBSERVICE command 122

ATTRLEN option (continued)
  CREATE TRANCLASS command 107
  CREATE TRANSACTION command 110
  CREATE TSMODEL command 113
  CREATE TYPETERM command 116
  CREATE UIMAP command 119
  CREATE WEBSERVICE command 122

AUDALARMST option
  INQUIRE TERMINAL command 363

AUDITLEVEL option
  INQUIRE PROCESSTYPE command 284, 559

AUDITLOG option
  INQUIRE PROCESSTYPE command 284

AUTHENTICATE option
  INQUIRE TCPIPSERVICE command 345
  AUTHORIZATION failures 16

AUTINSTMODEL option
  DISCARD AUTINSTMODEL command 127
  INQUIRE AUTINSTMODEL command 170

AUTINSTMODEL, DISCARD command 127

AUTINSTMODEL, INQUIRE command 170

AUTOCONNECT option
  INQUIRE CONNECTION command 187
  INQUIRE MODENAME command 270
  INQUIRE TERMINAL command 363

AUTOINSTALL option
  COLLECT STATISTICS command 41
  PERFORM STATISTICS command 467

AUTOINSTALL, INQUIRE command 171

AUTOINSTALL, SET command 474
  Automatic installation of terminals 171

AUTOPUBLISH option
  INQUIRE CORBASERVER command 194

AUTOSTARTST option
  INQUIRE CLASSCACHE command 180
  PERFORM CLASSCACHE command 443
  SET CLASSCACHE command 478

AUXSTATUS option
  INQUIRE TRACEDEST command 378
  SET TRACEDEST command 602

AVAILABILITY option
  INQUIRE DSNAME 223
  SET DSNAME 519

AVAILABLE option
  INQUIRE MODENAME command 270
  SET MODENAME command 552

B

BACKLOG option
  INQUIRE TCPIPSERVICE command 345
  SET TCPIPSERVICE command 585

BACKTRANSST option
  INQUIRE TERMINAL command 364

BACKUPTYPE option
  INQUIRE DSNAME command 223

BASEDSNAME option
  INQUIRE DSNAME command 224

INQUIRE FILE command 244

BATCH backout utility 222
BEAN option
  COLLECT STATISTICS command 42
  INQUIRE BEAN command 173
  PERFORM STATISTICS command 467
BEAN, INQUIRE command 173
BEANNNAME option
  INQUIRE REQUESTMODEL command 301
BINDING option
  INQUIRE WEBSERVICE command 437
BLOCKFORMAT option
  INQUIRE FILE command 244
  INQUIRE TDQUEUE command 352
BLOCKKEYLEN option
  INQUIRE FILE command 244
BLOCKSIZE option
  INQUIRE FILE command 244
  INQUIRE TDQUEUE command 352
BREXIT option
  INQUIRE TRANSACTION command 392
BRFACILITY option
  INQUIRE BRFACILITY command 175
  INQUIRE TASK command 330
BRFACILITY, INQUIRE command 175
BRFACILITY, SET command 476
BRIDGE option
  INQUIRE TASK command 331
BROWSE option
  INQUIRE FILE command 244
  SET FILE command 532
browsing
  AUTINSTMODEL entries 170
  BEAN entries 173
  BRFACILITY entries 175
  CFDTPOOL entries 178
  CONNECTION entries 186
  CORBASERVER entries 194
  DB2ENTRY entries 205
  DB2TRAN entries 210
  DJAR entries 217
  DOCTEMPLATE entries 219
  FILE entries 243, 259, 310
  JOURNALMODEL entries 257
  JVM entries 262
  JVM profiles 267
  MODENAME entries 269
  NETNAME entries 278
  PARTNER entries 280
  PIPELINES 282
  PROCESSSTYPE entries 284
  PROFILE entries 286
  PROGRAM entries 288
  REQUESTMODEL entries 301
  TDQUEUE entries 351
  TERMINAL entries 361
  TRANCLASS entries 386
  TRANSCOMPCODE entries 389
  TRANSACTION entries 392
  UOWs 408, 418, 423
  Workrequests 441
browsing resource definitions 21
browsing rules 23
BUSY option
  SET FILE command 532
C
C/370 language
  argument values 12
CACHEFREE option
  INQUIRE CLASSCACHE command 180
CACHESIZE option
  INQUIRE CLASSCACHE command 181
  PERFORM CLASSCACHE command 444
CALLER option
  PERFORM DUMP command 453
CALLERLENGTH option
  PERFORM DUMP command 453
CANCEL option
  SET CONNECTION command 483
CDSASIZE option
  INQUIRE SYSTEM command 319
CECI transaction 1
CEDF transaction 1
CEDFSTATUS option
  INQUIRE PROGRAM command 288
  SET PROGRAM command 561
CEMT transaction
  function provided by INQUIRE and SET
  commands 1
CERTIFICATE option
  INQUIRE CORBASERVER command 195
  INQUIRE TCPIPSERVICE command 346
  INQUIRE URIMAP command 428
CETR transaction
  function provided by INQUIRE and SET
  commands 1
CFDTPOOL option
  INQUIRE CFDTPOOL command 178
  INQUIRE FILE command 245
  SET FILE command 533
CFDTPOOL, INQUIRE command 178
char-expr argument, CICS command format 6
CHARACTERSET option
  INQUIRE URIMAP command 428
CICS value
  INQUIRE TCPIP command 342
CICS-supplied security 2
CICS-value data area (CVDA) 8
CICSSSTATUS option
  INQUIRE SYSTEM command 319
CICSSYS option
  INQUIRE SYSTEM command 320
CICSTSLEVEL option
  INQUIRE SYSTEM command 320
CIPEHRS option
  INQUIRE TCPIPSERVICE command 346
CIPHERS option
  INQUIRE CORBASERVER command 195
  INQUIRE URIMAP command 428
CLASSCACHE, INQUIRE command 180
CLASSCACHE, PERFORM command 443
CLASSCACHE, SET command 478
CLASSCACHEST option
  INQUIRE JVM command 263
  INQUIRE JVMPROFILE command 267
CLIENTAUTH value
  INQUIRE TCPIPSERVICE command 348
CLIENTCERT option
  INQUIRE CORBASERVER command 195
CLIENTIPADDR option
  INQUIRE WORKREQUEST command 441
CLOSED value
  INQUIRE TCPIP command 342
  INQUIRE TCPIPSERVICE command 347
  SET TCPIP command 584
  SET TCPIPSERVICE command 586
CLOSETIMEOUT option
  INQUIRE TCPIPSERVICE command 346
CLOSING value
  INQUIRE TCPIP command 342
  INQUIRE TCPIPSERVICE command 347
CMDPROTECT option
  INQUIRE SYSTEM command 320
CMDSEC option
  INQUIRE TASK command 331
  INQUIRE TRANSACTION command 392
COBOL
  argument values 11
COBOLTYPE option
  INQUIRE PROGRAM command 289
COLDSTATUS option
  INQUIRE SYSTEM command 320
COLLECT STATISTICS
  conditions 46
COLLECT STATISTICS command 38
COLORST option
  INQUIRE TERMINAL command 364
command interpreter transaction (CECI) 1
command security checking 16
command, CREATE FILE 69
commands
  format, arguments 2
COMPID option
  INQUIRE TRACETYPE command 383
  SET TRACETYPE command 608
COMPLETE option
  CREATE CONNECTION command 50
  CREATE TERMINAL command 106
CONCURRENCY option
  INQUIRE PROGRAM command 289
CONCURRENTST option
  INQUIRE EXITPROGRAM command 237
conditions
  ACQUIRE TERMINAL command 36
  COLLECT STATISTICS command 46
  CREATE CONNECTION command 50
  CREATE CORBASERVER command 53
  CREATE DB2ENTRY command 59
  CREATE DB2TRAN command 62
  CREATE DJAR command 64
  CREATE DOCTEMPLATE command 66
  CREATE ENQMODEL command 68
  CREATE FILE command 71
  CREATE JOURNALMODEL command 73
  CREATE LSRPOOL command 75
  CREATE MAPSET command 78
  CREATE PARTITIONSET command 80
  CREATE PARTNER command 82
  CREATE PIPELINE command 84
  CREATE PROCESTYPE command 86
  CREATE PROFILE command 88
  CREATE PROGRAM command 90
  CREATE REQUESTMODEL command 93
  CREATE SESSIONS command 95
  DISABLE PROGRAM command 125
  DISCARD AUTINSTMODEL command 127
  DISCARD CONNECTION command 129
  DISCARD CORBASERVER command 130
  DISCARD DB2ENTRY command 132
  DISCARD DJAR command 134
  DISCARD DOCTEMPLATE command 135
  DISCARD ENQMODEL command 136
  DISCARD FILE command 137
  DISCARD JOURNALMODEL command 138
  DISCARD JOURNALNAME command 140
  DISCARD PARTNER command 141
  DISCARD PIPELINE command 142
  DISCARD PROCESTYPE command 143
  DISCARD PROFILE command 144
  DISCARD PROGRAM command 145
  DISCARD REQUESTMODEL command 146
  DISCARD TCPIPSERVICE command 147
  DISCARD TDQUEUE command 148
  DISCARD TERMINAL command 150
  DISCARD TRANCLASS command 152
  DISCARD TRANSACTION command 153
  DISCARD TSMODEL command 154
  DISCARD URIMAP command 155
  DISCARD WEBSERVICE command 156
  ENABLE PROGRAM command 161
  EXTRACT EXIT command 165
  EXTRACT STATISTICS command 168
  INQUIRE AUTINSTMODEL command 170
  INQUIRE AUTOINSTALL command 172
  INQUIRE BEAN command 173
  INQUIRE BRFACILITY command 177
  INQUIRE CFDTPPOOL command 178
  INQUIRE CLASSCACHE command 182
  INQUIRE command 23
  INQUIRE CONNECTION command 193
  INQUIRE CORBASERVER command 196, 218
  INQUIRE DB2CONN command 204
  INQUIRE DB2ENTRY command 208
INQUIRE DB2TRAN command 211
INQUIRE DELETESHIPPED 213
INQUIRE DISPATCHER command 216
INQUIRE DOCTEMPLATE command 221
INQUIRE DSNAME command 227
INQUIRE DUMPDS command 230
INQUIRE ENQMODEL 233
INQUIRE EXCI command 234
INQUIRE EXITPROGRAM command 241
INQUIRE FILE command 252
INQUIRE HOST command 255
INQUIRE IRC command 256
INQUIRE JOURNALMODEL command 258
INQUIRE JOURNALNAME command 260
INQUIRE JVM command 264, 268
INQUIRE JVMPOOL command 266
INQUIRE MESSAGEX command 270, 311
INQUIRE MONITOR command 275
INQUIRE MVSTCB command 277
INQUIRE PARTY command 281
INQUIRE PIPELINE command 283
INQUIRE PROCESSTYPE command 285, 560
INQUIRE PROFILE command 286
INQUIRE PROFILE command 286
INQUIRE PROGRAM command 294
INQUIRE REQID command 299
INQUIRE REQUESTMODEL command 302
INQUIRE rmss command 303
INQUIRE STATISTICS command 306
INQUIRE STORAGE command 309
INQUIRE SUBPOOL 313
INQUIRE SYSDUMPCODE command 316
INQUIRE SYSTEM command 327
INQUIRE TASK command 338
INQUIRE TASK LIST command 340
INQUIRE TRANSACTION command 341
INQUIRE TCLASS command 341
INQUIRE TCPIP command 343
INQUIRE TCPIService command 348
INQUIRE TDOQUEUE command 357
INQUIRE TERMINAL command 376
INQUIRE TRACEDEST command 380
INQUIRE TRACETYPE command 382
INQUIRE TRANCLASS command 387
INQUIRE TRAN_DUMP command 390
INQUIRE TRANSACTION command 398
INQUIRE TSPOOL 401
INQUIRE TSQNAME 405
INQUIRE TSQUEUE 405
INQUIRE UOW command 411
INQUIRE UOWDSNF command 416
INQUIRE UOWENTRY command 422
INQUIRE UOWLINK command 425
INQUIRE URIMAP command 430
INQUIRE VOLUME command 432
INQUIRE VTAM command 435
INQUIRE WEB command 436
INQUIRE WEBSERVICE command 439
INQUIRE WORKREQUEST 442
PERFORM CLASSCACHE command 445
PERFORM CORBASERVER command 448
PERFORM DJAR command 451
PERFORM DUMP command 454
PERFORM ENDAFFINITY command 456
PERFORM PIPELINE command 458
PERFORM RESTART command 460
PERFORM SECURITY REBUILD command 461
PERFORM SHUTDOWN command 464
PERFORM STATISTICS RECORD command 469
RESYNC ENTRYNAME command 473
SET AUTOINSTALL command 475
SET BRIEFCOMMAND 476
SET CLASSCACHE command 479
SET CONNECTION command 486
SET CORBASERVER command 490
SET DB2CONN command 499
SET DB2ENTRY command 506
SET DB2TRAN command 508
SET DELETESHIPPED command 511
SET DISPATCHER command 515
SET DSNAME command 521
SET DUMPDS command 526
SET ENQMODEL command 527
SET FILE command 538
SET HOST command 543
SET IRC command 544
SET JOURNALNAME command 547
SET JVMPOOL command 550
SET MESSAGEX command 552
SET MONITOR command 556
SET NETNAME command 557
SET PIPELINE command 558
SET PROGRAM command 564
SET STATISTICS command 568
SET SYSDUMPCODE command 572
SET SYSTEM command 578
SET TASK command 581
SET TCPIP command 584
SET TCPIService command 586
SET TDOQUEUE command 591
SET TERMINAL command 600
SET TRACEDEST command 604
SET TRACETYPE command 607
SET TRACETYPE command 610
SET TRACETYPE command 612
SET TRANCLASS command 612
SET TRANCLASS command 615
SET TRANSACTION command 619
SET TRANSACTION command 619
SET TSQNAME command 621
SET TSQUEUE command 621
SET UOW command 623
SET UOWLINK command 625
SET URIMAP command 628
SET VOLUME command 629
SET VTAM command 632
SET WEBSERVICE command 634
SET WORKREQUEST command 636
SET CONNECTION command 481
CONNECTION option
  COLLECT STATISTICS command 42
  CREATE CONNECTION command 50
  INQUIRE CONNECTION command 188
  INQUIRE MODENAME command 270
  PERFORM STATISTICS command 467
  SET MODENAME command 552
CONNECTION, CREATE command 48
CONNECTION, DISCARD command 128
CONNECTION, INQUIRE command 183
CONNECTION, SET command 480, 625
CONNECTIONS option
  INQUIRE TCPIPSERVICE command 346
CONNECTST option
  INQUIRE EXITPROGRAM command 237
CONNSTATUS option
  INQUIRE CFDTPOOL command 178
  INQUIRE CONNECTION command 188
  INQUIRE TSPool command 401
  SET CONNECTION command 481
CONNTYPE option
  INQUIRE CONNECTION command 189
CONSOLE option
  INQUIRE TERMINAL command 364
CONSOLES option
  INQUIRE AUTOINSTALL command 171
  SET AUTOINSTALL command 474
CONTAINER option
  INQUIRE WEBSERVICE command 437
CONVERSEST option
  INQUIRE MONITOR command 273
  SET MONITOR command 554
CONVERTER option
  INQUIRE URIMAP command 428
COPID
  option of DSNCRCT macro 495
COPY option
  INQUIRE PROGRAM command 290
  SET PROGRAM command 561
COPYST option
  INQUIRE TERMINAL command 364
CORBASERVER option
  COLLECT STATISTICS command 42
  CREATE CORBASERVER command 53
  INQUIRE BEAN command 173
  INQUIRE CORBASERVER command 195
  INQUIRE REQUESTMODEL command 301
  INQUIRE WORKREQUEST command 441
  PERFORM CORBASERVER command 447
  PERFORM STATISTICS command 467
CORBASERVER, CREATE command 52
CORBASERVER, DISCARD command 130
CORBASERVER, INQUIRE command 194
CORBASERVER, PERFORM 447
CORBASERVER, SET command 489
CORBASERVER option
  INQUIRE DJAR command 217
  PERFORM DJAR command 451
  SET CORBASERVER command 489
CORRELID option
  INQUIRE TERMINAL command 364
  CREATE CONNECTION command 48
  conditions 50
  CREATE CORBASERVER command 52
  conditions 53
  CREATE DB2CONN command 55
  CREATE DB2ENTRY command 58
  conditions 59
  CREATE DB2TRAN command 61
  conditions 62
  CREATE DJAR command 63
  conditions 64
  CREATE DOCTEMPLATE command 65
  conditions 66
  CREATE ENQMODEL command 67
  conditions 68
  CREATE FILE command 69
  conditions 71
  CREATE JOURNALMODEL command 72
  conditions 73
  CREATE LSRPOOL command 74
  conditions 75
  CREATE MAPSET command 77
  conditions 78
  CREATE PARTITIONSET command 79
  conditions 80
  CREATE PARTNER command 81
  conditions 82
  CREATE PIPELINE command 83
  conditions 84
  CREATE PROCSSTYPE command 85
  conditions 86
  CREATE PROFILE command 87
  conditions 88
  CREATE PROGRAM command 89
  conditions 90
  CREATE REQUESTMODEL command 92
  conditions 93
  CREATE SESSIONS command 94
  conditions 95
  CREATE TCPIPSERVICE command 97
  conditions 98
  CREATE TDQUEUE command 100
  conditions 102
  CREATE TERMINAL command 104
  conditions 106
  CREATE TRANCLASS command 107
  conditions 108
  CREATE TRANSACTION command 109
  conditions 110
  CREATE TSMODEL command 112
  conditions 113
  CREATE TYPETERM command 114
  conditions 116
  CREATE URIMAP command 118
  conditions 119
  CREATE WEBSERVICE command 121
  conditions 122
CREATESESS option
  INQUIRE TERMINAL command 364
  SET TERMINAL command 595
creating resource definitions 24
CRITICAL value 346
CRLPROFILE option
   INQUIRE TCPIP command 342
CTERM option
   DSNCRCT macro 495
CTX option of DSNCRCT macro 495
CURAUXDS option
   INQUIRE TRACEDEST command 378
CURRENT option
   INQUIRE SYSDUMPCODE command 314
   INQUIRE TCLASS command 341
   INQUIRE TRANDUMPCODE command 389
CURRENTDDS option
   INQUIRE DUMPDS command 229
CURRENTPROG option
   INQUIRE TASK command 331
CURREQS option
   INQUIRE AUTOINSTALL command 172
CVDA (CICS-value data area)
   argument values 6
   command format 6
   example code 9
   listed in numerical and alphabetical order 637
   on INQUIRE commands 8
CVDA values
   CLIENT
      INQUIRE URIMAP command 430
   DISABLED
      INQUIRE HOST command 254
      INQUIRE URIMAP command 428, 429
      SET HOST command 542
      SET URIMAP command 627
   DISABLEDHOST
      INQUIRE URIMAP command 428
   ENABLED
      INQUIRE HOST command 254
      INQUIRE URIMAP command 428, 429
      SET HOST command 542
      SET URIMAP command 627
   NO
      INQUIRE URIMAP command 428
   NONE
      INQUIRE URIMAP command 430
      SET URIMAP command 628
   PERM
      INQUIRE URIMAP command 430
   PERMANENT
      SET URIMAP command 628
   PIPELINE
      INQUIRE URIMAP command 430
   SERVER
      INQUIRE URIMAP command 430
   TEMP
      INQUIRE URIMAP command 430
   TEMPORARY
      SET URIMAP command 628
   YES
      INQUIRE URIMAP command 428
D
DAEOPTION option
   INQUIRE SYSDUMPCODE command 315, 571
   data table options
      MAXNUMRECS option on SET FILE command 535
      TABLE option on SET FILE command 537
   data types 11
   data-area argument
      CICS command format 6
   data-areas 6
   data-value argument
      CICS command format 6
      data-values 6
   DATABASES option
      INQUIRE TDQUEUE command 352
   DATALOCATION option
      INQUIRE PROGRAM command 290
   DATASTREAM option
      INQUIRE TERMINAL command 365
   DB2 option
      PERFORM STATISTICS command 467
   DB2CONN command
      conditions 56
   DB2CONN option
      COLLECT STATISTICS command 42
      DISCARD DB2CONN command 131
      INQUIRE SYSTEM command 320
      DB2CONN, CREATE command 55
      DB2CONN, DISCARD command 131
      DB2CONN, INQUIRE command 197
      DB2CONN, SET command 492
   DB2ENTRY option
      COLLECT STATISTICS command 42
      DISCARD DB2ENTRY command 132
      DB2ENTRY, CREATE command 58
      DB2ENTRY, DISCARD command 132
      DB2ENTRY, INQUIRE command 205
      DB2ENTRY, SET command 502
   DB2PLAN option
      INQUIRE TASK command 331
   DB2TRAN option
      DISCARD DB2TRAN command 133
      DB2TRAN, CREATE command 61
      DB2TRAN, DISCARD command 133
      DB2TRAN, INQUIRE command 210
      DB2TRAN, SET command 508
   DDNAME option
      INQUIRE DOCTEMPLATE command 220
      INQUIRE TDQUEUE command 352
   DEBUGTOOL option
      INQUIRE SYSTEM command 320
      SET SYSTEM command 575
   defining exits 28
   DEFPAGEHT option
      INQUIRE TERMINAL command 365
   DEFPAGEWD option
      INQUIRE TERMINAL command 365
   DEFS_CRNHT option
      INQUIRE TERMINAL command 365
   DEFS_CRNWD option
      INQUIRE TERMINAL command 365
Index 711
DELETE option
  INQUIRE FILE command 245
  SET FILE command 533
DELETESHIPPED, INQUIRE command 212
DELETESHIPPED, PERFORM command 450
DELETESHIPPED, SET command 510
DEREGERROR value 346
DEREGISTERED option
  SET VTAM command 630
DEREGISTERED value 346
DEVICE option
  INQUIRE TERMINAL command 365
DFLTUSER option
  INQUIRE SYSTEM command 321
DISABLE PROGRAM command 123
  conditions 125
examples for global user exits 126
DISABLED CVDA value
  INQUIRE AUTOINSTALL command 172
DISCARD AUTINSTMODEL command 127
  conditions 127
DISCARD commands
  CONNECTION 128
  CORBASERVER 130
  DJAR 134
  PIPELINE 142
  TERMINAL 150
  WEBSERVICE 156
DISCARD CONNECTION command 128
  conditions 129
DISCARD CORBASERVER command 130
  conditions 130
DISCARD DB2CONN command 131
  conditions 131
DISCARD DB2ENTRY command 132
  conditions 132
DISCARD DB2TRAN command 133
  conditions 133
DISCARD DJAR command 134
  conditions 134
DISCARD DOCTEMPLATE command 135
  conditions 135
DISCARD ENQMODEL command 136
  conditions 136
DISCARD FILE command 137
  conditions 137
DISCARD JOURNALMODEL command 138
  conditions 138
DISCARD JOURNALNAME command 139
  conditions 140
DISCARD option
  CREATE CONNECTION command 50
  CREATE TERMINAL command 106
DISCARD PARTNER command 141
  conditions 141
DISCARD PIPELINE command 142
  conditions 142
DISCARD PROCESSTYPE command 143
  conditions 143
DISCARD PROFILE command 144
  conditions 144
DISCARD PROGRAM command 145
  conditions 145
DISCARD REQUESTMODEL command 146
  conditions 146
DISCARD TCPIPSERVICE command 147
  conditions 147
DISCARD TDQUEUE command 148
  conditions 148
DISCARD TERMINAL command 150
  conditions 150
DISCARD TRANCLASS command 152
  conditions 152
DISCARD TRANSACTION command 153
  conditions 153
DISCARD TSMODEL command 154
  conditions 154
DISCARD URMIP command 155
  conditions 155
DISCARD WEBSERVICE command 156
  conditions 156
discarding resources
  resource definitions 27, 127
DISCREQST option
  INQUIRE TERMINAL 595
  INQUIRE TERMINAL command 365
DISPATCHABLE option
  INQUIRE TASK LIST command 339
DISPATCHER option
  COLLECT STATISTICS command 42
  PERFORM STATISTICS command 467
DISPATCHER, INQUIRE command 214
DISPATCHER, SET command 513
DISPOSITION option
  INQUIRE BEAN command 173
  INQUIRE DJAR command 217
  DJAR, CREATE command 63
  DJAR, DISCARD command 134
  DJAR, INQUIRE command 217
  DJAR, PERFORM 451
DJARDIR option
  INQUIRE CORBASERVER command 195
DNSGROUP option
  INQUIRE TCPIPSERVICE command 346
DNSSTATUS option
  INQUIRE TCPIPSERVICE command 346
  SET TCPIPSERVICE command 585
DOCTEMPLATE option
  INQUIRE DOCTEMPLATE command 220
DOCTEMPLATE, DISCARD command 135
DOCTEMPLATE, INQUIRE command 219
DSALIMIT option
  INQUIRE SYSTEM command 321
  SET SYSTEM command 575
DSANAME option
  INQUIRE STORAGE command 308
DSANAME option (continued)
INQUIRE SUBPOOL command 312
DSNAME option
INQUIRE DOCTEMPLATE command 220
INQUIRE DSNAME command 224
INQUIRE FILE command 245
INQUIRE TDQUEUE command 353
SET DSNAME command 519
SET FILE command 533
DSNAME, INQUIRE command 222
DSNAME, SET command 516
DSRTPROGRAM option
INQUIRE SYSTEM command 321
SET SYSTEM command 575
DTIMEOUT option
INQUIRE TASK command 331
INQUIRE TRANSACTION command 392
DTRPROGRAM option
INQUIRE SYSTEM command 321
SET SYSTEM command 575
DUALCASEST option
INQUIRE TERMINAL command 365
dump data sets 229
DUMP option
PERFORM SHUTDOWN command 463
DUMP, PERFORM command 453
DUMPCODE option
PERFORM DUMP command 453
DUMPDS, INQUIRE command 229
DUMPDS, SET command 525
DUMPID option
PERFORM DUMP command 454
DUMPING option
INQUIRE SYSTEM command 321
INQUIRE TASK command 331
INQUIRE TRANSACTION command 392
SET SYSTEM command 575
SET TRANSACTION command 617
DUMPSCOPE option
INQUIRE SYSDUMPCODE command 315, 571
INQUIRE TRANDUMPCODE command 389
SET TRANDUMPCODE command 614
DURATION option
INQUIRE UOWENQ command 419
DYNAMSTATUS option
INQUIRE PROGRAM command 290

E

ECDSASIZE option
INQUIRE SYSTEM command 321
ECI value
INQUIRE TCPIPSERVICE command 347
EDSALIMIT option
INQUIRE SYSTEM command 321
SET SYSTEM command 576
EDSASIZE option
INQUIRE SYSTEM command 322
ELEMENT option
INQUIRE STORAGE command 308
ELEMENTLIST option
INQUIRE MVSTCB command 276
INQUIRE STORAGE command 308
EMPTY option
SET FILE command 534
EMPTYSTATUS option
INQUIRE FILE command 245
INQUIRE TDQUEUE command 353
SET FILE command 534
ENABLE PROGRAM command 157
conditions 161
examples for global user exits 162
examples for task-related user exits 163
ENABLED CVDA value
INQUIRE AUTOINSTALL command 172
ENABLESTATUS option
INQUIRE AUTOINSTALL command 172
INQUIRE CORBASERVER command 195
INQUIRE FILE command 245
INQUIRE HOST command 254
INQUIRE TDQUEUE command 353
INQUIRE URIMAP command 428
SET CORBASERVER command 489
SET FILE command 534
SET HOST command 542
SET TDQUEUE command 590
SET URIMAP command 627
END condition
INQUIRE HOST command 255
INQUIRE URIMAP command 430
ENDAFFINITY, PERFORM command 456
ENDOFDAY option
INQUIRE STATISTICS command 305
SET STATISTICS command 567
ENDOFDAYHRS option
INQUIRE STATISTICS command 305
SET STATISTICS command 567
ENDOFDAYMINS option
INQUIRE STATISTICS command 305
SET STATISTICS command 567
ENDOFDAYSECS option
INQUIRE STATISTICS command 305
SET STATISTICS command 567
ENDPOINT option
INQUIRE WEBSERVICE command 437
ENQ, INQUIRE command 231
ENQFAILS option
INQUIRE UOWENQ command 419
ENQMODEL option
CREATE ENQMODEL command 68
DISCARD ENQMODEL command 136
INQUIRE ENQMODEL command 232
ENQMODEL, DISCARD command 136
ENQMODEL, INQUIRE command 232
ENQNAME option
INQUIRE ENQMODEL command 232
ENQSCOPE option
INQUIRE ENQMODEL command 232
INQUIRE UOWENQ command 419
ENQUEUE option
COLLECT STATISTICS command 42
ENQUEUE option (continued)
PERFORM STATISTICS command 467
ENTRY option
ENABLE PROGRAM command 158
INQUIRE EXITPROGRAM command 238
ENTRYNAME option
DISABLE PROGRAM command 123
ENABLE PROGRAM command 158
EXTRACT EXIT command 164
INQUIRE EXITPROGRAM command 238
RESYNC command 472
ENTRYNAME, RESYNC command 471
ENTRYPOINT option
INQUIRE PROGRAM command 158
ERDSASIZE option
INQUIRE SYSTEM command 322
ERROROPTION option
INQUIRE TDQUEUE command 353
ESM (external security manager) 2
ESMRESP option
PERFORM SECURITY REBUILD command 461
EUDSASIZE option
INQUIRE SYSTEM command 322
Examples
using the INQUIRE BEAN command 174
using the SET PROGRAM command 565
EXCEPTCLASS option
INQUIRE MONITOR command 273
SET MONITOR command 554
EXCI option
INQUIRE EXCI command 234
EXCI, INQUIRE command 234
EXCLUSIVE option
INQUIRE FILE command 246
SET FILE command 535
EXEC CICS commands
format 2
Function codes 671
Response Codes 671
EXEC CICS CREATE
RESP2 values 681
EXEC CICS SPI commands
Threadsafe 687
EXCEKEY option
INQUIRE JVM command 263
INQUIRE PROGRAM command 291
execution diagnostic facility transaction (CEDF) 1
EXECUTIONSET option
INQUIRE PROGRAM command 291
SET PROGRAM command 562
exit names 29
EXIT option
DISABLE PROGRAM command 124
ENABLE PROGRAM command 158
INQUIRE DOCTEMPLATE command 220
INQUIRE EXITPROGRAM command 238
exit-related commands 28
EXIT, EXTRACT command 164
EXITALL option
DISABLE PROGRAM command 124
EXITPGM option
INQUIRE DOCTEMPLATE command 220
EXITPROGRAM option
INQUIRE EXITPROGRAM command 238
EXITPROGRAM, INQUIRE command 236
exits
defining 28
EXITTRACING option
INQUIRE CONNECTION command 189
INQUIRE TERMINAL command 366
SET CONNECTION command 482
SET NETNAME command 557
SET TERMINAL command 595
EXTENDEDSSID option
INQUIRE TERMINAL command 366
external security manager (ESM) 2
EXTRACT EXIT command 164
conditions 165
EXTRACT STATISTICS
conditions 168
EXTRACT STATISTICS command 166

F
FACILITY option
INQUIRE TASK command 331
FACILITYLIKE option
INQUIRE TRANSACTION command 393
FACILITYTYPE option
INQUIRE TASK command 331
FEPI option
PERFORM STATISTICS command 467
FILE option
COLLECT STATISTICS command 42
CREATE FILE command 71
DISCARD FILE command 137
INQUIRE DOCTEMPLATE command 220
INQUIRE FILE command 246
INQUIRE PROCESSTYPE command 284
PERFORM STATISTICS command 467
SET FILE command 535
FILE, DISCARD command 137
FILE, INQUIRE command 242
FILE, SET command 529
FILECOUNT option
INQUIRE DSNAME command 224
FILELIMIT option
INQUIRE MONITOR command 273
filename argument, CICS command format 6
FLENGTH option
INQUIRE STORAGE command 308
INQUIRE TSONAME command 404
INQUIRE TSQUEUE command 404
FMHPARMST option
INQUIRE TERMINAL command 366
FMHSTATUS option
INQUIRE REQID command 297
FORCECANCEL option
SET CONNECTION command 483
FORCEQR option
INQUIRE SYSTEM command 322
FORCEQR option  (continued)
  SET SYSTEM command  576
format rules  5
FORMATEDF option
  DISABLE PROGRAM command  124
  ENABLE PROGRAM command  158
FORMATEDFST option
  INQUIRE EXITPROGRAM command  238
FORMFEEDST option
  INQUIRE TERMINAL command  366
FREQUENCY option
  INQUIRE MONITOR command  273
  SET MONITOR command  555
FREQUENCYHRS option
  INQUIRE MONITOR command  273
  SET MONITOR command  555
FREQUENCYMIN option
  INQUIRE MONITOR command  273
  SET MONITOR command  555
FREQUENCYSEC option
  INQUIRE MONITOR command  274
  SET MONITOR command  555
Function codes
  of EXEC CICS commands  671
  function shipping, not available for SP commands  1
FWDRECOVLOG option
  INQUIRE DSNAME  224
FWDRECOVLSN option
  INQUIRE DSNAME  224
FWDRECASTATUS option
  INQUIRE FILE command  246

G
GENTRYNAME option
  ENABLE PROGRAM command  158
  INQUIRE EXITPROGRAM command  239
GALength option
  ENABLE PROGRAM command  159
  EXTRACT EXIT command  164
  INQUIRE EXITPROGRAM command  239
GARBAGEINT
  CEMT INQUIRE WEB  436
  SET WEB command  633
GASET option
  EXTRACT EXIT command  164
GAUSECOUNT option
  INQUIRE EXITPROGRAM command  239
GCHARS option
  INQUIRE TERMINAL command  366
GCODES option
  INQUIRE TERMINAL command  367
GMMLLENGTH option
  INQUIRE SYSTEM command  322
  SET SYSTEM command  576
GMTEXT option
  INQUIRE SYSTEM command  322
  SET SYSTEM command  576
GMTRANID option
  INQUIRE SYSTEM command  322
GRNAME option
  INQUIRE CONNECTION command  189
  INQUIRE VTAM command  433
GROUP option
  DSNCRTCT macro  503
GRPCRITICAL option
  INQUIRE TCPIPSERVICE command  346
GRSTATUS option
  INQUIRE VTAM command  433
GTFSSTATUS option
  INQUIRE TRACEDEST command  379
  SET TRACEDEST command  603

H
HFORMST option
  INQUIRE TERMINAL command  367
HFSFILE option
  INQUIRE DOCTEMPLATE command  220
  INQUIRE ULRMP command  428
HFSFLR option
  INQUIRE DJAR command  217
HFSNAME option
  INQUIRE JVMPROFILE command  267
HIGHLIGHTST option
  INQUIRE TERMINAL command  367
HOLDSTATUS option
  INQUIRE PROGRAM command  291
HOST option
  INQUIRE CORBASERVER command  196
  INQUIRE HOST command  254, 542
  INQUIRE UOWLINK command  424
  INQUIRE ULRMP command  428
HOSTCODEPAGE option
  INQUIRE ULRMP command  428
HOURS option
  INQUIRE REQID command  298
HTTP value
  INQUIRE TCPIPSERVICE command  347

I
IDENTIFIER option
  INQUIRE TASK command  332
IDLIST option
  RESYNC command  472
IDLISTLENGTH option
  RESYNC command  472
IGNORE (null values)  14
IIOP value
  INQUIRE TCPIPSERVICE command  347
ILLOGIC condition
  INQUIRE HOST command  255
  INQUIRE ULRMP command  430
IMMCLOSE value
  SET TCPIP command  584
  SET TCPIPSERVICE command  586
IMMCLOSING value
  INQUIRE TCPIP command  342
  INQUIRE TCPIPSERVICE command  347
IMMEDIATE option
   PERFORM SHUTDOWN command  463
INDIRECTNAME option
   INQUIRE TDQUEUE command  353
INDOUBT option
   INQUIRE TASK command  332
   INQUIRE TDQUEUE command  354
   INQUIRE TRANSACTION command  393
INDOUBTMINS option
   INQUIRE TASK command  332
   INQUIRE TRANSACTION command  393
INDOUBTST option
   INQUIRE EXITPROGRAM command  239
INDOUBTWAIT option
   ENABLE PROGRAM command  159
   INQUIRE TDQUEUE command  354
   INQUIRE TRANSACTION command  332, 393
INITIALDDS option
   INQUIRE DUMPDS command  229
   SET DUMPDS command  525
INITIALIZE option
   PERFORM CLASSCACHE command  444
INITSSTATUS option
   INQUIRE SYSTEM command  322
INQUIRE and SET commands
   examples
      Assembler  11
      C  10
      COBOL  10
      PL/I  10
   null values  14
INQUIRE AUTINSTMODEL command  170
   conditions  170
INQUIRE AUTOINSTALL command  171
   conditions  172
INQUIRE BEAN command  173
   conditions  173
INQUIRE BRFA CILITY command  175
   conditions  177
INQUIRE CFDTPOOL command  178
INQUIRE CLASSCACHE command  180
   conditions  182
INQUIRE command, browse
   conditions  23
INQUIRE commands
   AUTINSTMODEL  170
   AUTOINSTALL  171
   BEAN  173
   BRFA CILITY  175
   CFDTPOOL  178
   CLASSCACHE  180
   CONNECTION  183
   CORBASERVER  194
   DB2CONN  197
   DB2ENTRY  205
   DB2TRAN  210
   DELETESHIPPED  212
   DISPATCHER  214
   DJAR  217
   DOCTEMPLATE  219
   DSNNAME  222
INQUIRE WORKREQUEST command
  conditions 442
  WORKREQUEST 440
inquiry commands 19
integer-expr argument, CICS command format 6
INTERFACE option
  INQUIRE REQUESTMODEL command 301
INTERVAL option
  INQUIRE REQID command 298
  INQUIRE STATISTICS command 305
  SET STATISTICS command 568
INTERVALHRS option
  INQUIRE STATISTICS command 305
  SET STATISTICS command 568
INTERVALMINS option
  INQUIRE STATISTICS command 305
  SET STATISTICS command 568
INTERVALSECS option
  INQUIRE STATISTICS command 305
  SET STATISTICS command 568
INTFACETYPE option
  INQUIRE REQUESTMODEL command 301
INTSTATUS option
  INQUIRE TRACEDEST command 379
  SET TRACEDEST command 603
INVREQ condition
  DISCARD URIMAP command 155
  INQUIRE HOST command 255
  SET URIMAP command 628
  SET WEBSERVICE command 634
IOTYPE option
  INQUIRE TDQUEUE command 354
IRC, INQUIRE command 256
IRC, SET command 544
ISOLATEST option
  INQUIRE TASK command 333
  INQUIRE TRANSACTION command 394

J
JNDIPREFIX option
  INQUIRE CORBASERVER command 196
JOBNAME option
  INQUIRE SYSTEM command 323
JOURNALMODEL option
  CREATE JOURNALMODEL command 73
  JOURNALMODEL, CREATE command 72
  JOURNALMODEL, DISCARD command 138
  JOURNALMODEL, INQUIRE command 257
JOURNALNAME option
  COLLECT STATISTICS command 42
  PERFORM STATISTICS command 467
  JOURNALNAME, DISCARD command 139
  JOURNALNAME, INQUIRE command 259
  JOURNALNAME, SET command 546
JOURNALNUM option
  COLLECT STATISTICS command 42
  INQUIRE FILE command 246
  PERFORM STATISTICS command 468
  JOURNALNUM, INQUIRE command 261
  JOURNALNUM, SET command 548
JVM, INQUIRE command 262
JVMCLASS option
  INQUIRE PROGRAM command 291
  SET PROGRAM command 562
JVMDEBUG option
  INQUIRE PROGRAM command 291
JVMLVLnTRACE option
  INQUIRE JVMPOOL command 265
  SET JVMPOOL command 549
JVMPOOL option
  COLLECT STATISTICS command 43
  PERFORM STATISTICS command 468
  JVMPOOL, INQUIRE command 265
  JVMPOOL, SET command 549
JVMPROFILE option
  COLLECT STATISTICS command 43
  INQUIRE PROGRAM command 291
  PERFORM STATISTICS command 468
  SET PROGRAM command 562
  JVMPROFILE, INQUIRE command 267
  JVMPROFILE, SET command 549
JVMUSERTRACE option
  INQUIRE JVMPOOL command 265
  SET JVMPOOL command 549

K
KATAKANAST option
  INQUIRE TERMINAL command 367
KEEPTIME option
  INQUIRE BRFACILITY command 175
KEYLENGTH option
  INQUIRE FILE command 247
  SET FILE command 535
KEYPOSITION option
  INQUIRE FILE command 247

L
label argument, CICS command format 6
LANGDEPRECATED option
  INQUIRE PROGRAM command 291
LANGUAGE option
  INQUIRE PROGRAM command 292
LASTMODTIME option
  INQUIRE DJAR command 217
  INQUIRE WEBSERVICE command 438
LASTRESET option
  COLLECT STATISTICS command 43
  EXTRACT STATISTICS command 167
  LASTRESETHRS option
  COLLECT STATISTICS command 43
  EXTRACT STATISTICS command 168
  LASTRESETMIN option
  COLLECT STATISTICS command 43
  EXTRACT STATISTICS command 168
  LASTRESETSEC option
  COLLECT STATISTICS command 43
  EXTRACT STATISTICS command 168
LASTUSEDINT option
INQUIRE TSQNAME command 404
INQUIRE TSQUEUE command 404
LENGTH option
default (PL/I) 11
INQUIRE PROGRAM command 292
INQUIRE REQID command 298
LENGTHLIST option
INQUIRE MVSTCB command 276
INQUIRE STORAGE command 308
LIGHTPENST option
INQUIRE TERMINAL command 367
LINK option
INQUIRE UOW command 408
INQUIRE UOWLINK command 424
LINKEDITMODE option
ENABLE PROGRAM command 159
LINKSYSNET option
INQUIRE BRFACILITY command 176
LINKSYSTEM option
INQUIRE BRFACILITY command 176
INQUIRE CONNECTION command 190
INQUIRE TERMINAL command 367
LISTENERPORT option
INQUIRE WORKREQUEST command 441
LISTSIZE option
INQUIRE TASK LIST command 339
literal constants 13
LOADPOINT option
INQUIRE PROGRAM command 292
LOADTYPE option
INQUIRE FILE command 247
SET FILE command 535
LOCATION option
INQUIRE TSQNAME command 404
INQUIRE TSQUEUE command 404
INQUIRE URIMAP command 429
SET URIMAP command 627
LOG CVDA value
CREATE CONNECTION command 50
CREATE CORBASERVER command 53
CREATE DB2CONN command 56
CREATE DB2ENTRY command 59
CREATE DB2TRAN command 62
CREATE DOCTEMPLATE command 66
CREATE ENOMODEL command 68
CREATE FILE command 71
CREATE JOURNALMODEL command 73
CREATE LSRPOOL command 75
CREATE MAPSET command 78
CREATE PARTITIONSET command 80
CREATE PARTNER command 82
CREATE PROCESSTYPE command 86
CREATE PROFILE command 88
CREATE PROGRAM command 90
CREATE REQUESTMODEL command 93
CREATE SESSIONS command 95
CREATE TCPIPSERVICE command 98
CREATE TCQUEUE command 102
CREATE TERMINAL command 106
CREATE TRANCCLASS command 108
LOG CVDA value (continued)
CREATE TRANSACTION command 110
CREATE TSMODEL command 113
CREATE TYPETERM command 116
CREATE URI MAP command 119
LOGDEFER option
INQUIRE SYSTEM command 323
SET SYSTEM command 576
LOSTLOCKS option
INQUIRE DSNAME 225
LPASTATUS option
INQUIRE PROGRAM command 293
LSRPOOL option
COLLECT STATISTICS command 43
CREATE LSRPOOL command 75
PERFORM STATISTICS command 468
LSRPOOL, CREATE command 74
LSRPPOOLID option
INQUIRE FILE command 248
SET FILE command 535

M
MAPNAME option
INQUIRE TERMINAL command 367
SET TERMINAL command 595
MAPSET option
CREATE MAPSET command 78
MAPSET, CREATE command 77
MAPSETNAME option
INQUIRE TERMINAL command 368
SET TERMINAL command 595
MAXACTIVE option
INQUIRE TRANCCLASS command 386
SET TRANCCLASS command 611
MAXDATALENV option
INQUIRE TCPIPSERVICE command 347
SET TCPIPSERVICE command 586
MAXIMUM option
INQUIRE MODENAME command 270
INQUIRE SYSDUMPCODE command 315
INQUIRE TCLASS command 341
INQUIRE TRANDUMPCODE command 389
SET SYSDUMPCODE command 571
SET TCLASS command 582
SET TRANDUMPCODE command 614
MAXITEMLEN option
INQUIRE TSQNAME command 404
INQUIRE TSQUEUE command 404
MAXJVMTCBS option
INQUIRE DISPATCHER command 215
SET DISPATCHER command 513
MAXNUMRECS option
INQUIRE FILE command 248
SET FILE command 535
MAXOPENTCBS option
INQUIRE DISPATCHER command 215
INQUIRE SYSTEM command 323
SET DISPATCHER command 513
SET SYSTEM command 576
NOLOG CVDA value (continued)
CREATE TRANCLASS command 108
CREATE TRANSACTION command 110
CREATE TSMODEL command 113
CREATE TYPETERM command 116
CREATE URIMAP command 119
NONCRITICAL value 346
NOQUEUE option
ACQUIRE TERMINAL command 36
NORESTART option
PERFORM SHUTDOWN command 463
NOSDTRAN option
PERFORM SHUTDOWN command 464
NOSSL value
INQUIRE TCPIPSERVICE command 348
NOTAPPLIC 14
NOTAPPLIC value 346
NOTAUTH condition 16
DISCARD URIMAP command 155
INQUIRE HOST command 255
INQUIRE URIMAP command 431
INQUIRE WEBSERVICE command 439
SET HOST command 543
SET URIMAP command 628
SET WEBSERVICE command 634
NOTFND condition
DISCARD URIMAP command 155
INQUIRE HOST command 255
INQUIRE URIMAP command 431
INQUIRE WEBSERVICE command 439
SET HOST command 543
SET URIMAP command 628
SET WEBSERVICE command 634
NOTSUPPORTED value
INQUIRE TCPIPSERVICE command 347
NQNAME option
INQUIRE CONNECTION command 190
INQUIRE TERMINAL command 369
null values 14
NUMCIPHER option
INQUIRE CORBASERVER command 196
INQUIRE TCPIPSERVICE command 347
NUMCIPHERS option
INQUIRE URIMAP command 429
NUMELEMENTS option
INQUIRE MVSTCB command 276
INQUIRE STORAGE command 308
NUMEXITS option
INQUIRE EXITPROGRAM command 239
NUMITEMS option
INQUIRE TDQUEUE command 355
INQUIRE TSQNAME command 405
INQUIRE TSQUEUE command 405

OBJECT option
INQUIRE DSNNAME command 225
INQUIRE FILE command 248
OBPERIDST option
INQUIRE TERMINAL command 370
OLDCACHES option
INQUIRE CLASSCACHE command 181
OPEN value
INQUIRE TCPIP command 342
INQUIRE TCPIPSERVICE command 347
SET TCPIP command 583
SET TCPIPSERVICE command 586
OPENSTATUS option
INQUIRE DUMPDS command 229
INQUIRE FILE command 248
INQUIRE IRC command 256
INQUIRE RMFS command 303
INQUIRE TCPIP command 342
INQUIRE TDQUEUE command 355
INQUIRE VTAM command 434
SET DUMPDS command 525
SET FILE command 536
SET IRC command 544
SET TCPIP command 583
SET TCPIPSERVICE command 586
SET TDQUEUE command 590
SET VTAM command 630
OPERATION option
INQUIRE REQUESTMODEL command 302
OPERID option
INQUIRE TERMINAL command 370
SET TERMINAL command 596
OPID
option of DSNCRCT macro 199, 504
OPREL option
INQUIRE SYSTEM command 323
OPSYS option
INQUIRE SYSTEM command 323
OSLEVEL
CEMT INQUIRE SYSTEM 323
OTASK option
INQUIRE WORKREQUEST command 441
OTTRANSID option
INQUIRE WORKREQUEST command 441
OTSTID option
INQUIRE UOW command 408
OTSTIMEOUT option
INQUIRE TRANSACTION command 394, 618
OUTLINEST option
INQUIRE TERMINAL command 370
OUTPRIVACY option
INQUIRE CORBASERVER command 196

P
PAGEHT option
INQUIRE TERMINAL command 370
PAGESTATUS option
INQUIRE TERMINAL command 370
SET TERMINAL command 596
PAGEWD option
  INQUIRE TERMINAL command 370
PARTIAL option
  RESYNC command 472
PARTITIONSET option
  CREATE PARTITIONSET command 80
PARTITIONSET, CREATE command 79
PARTITIONSSST option
  INQUIRE TERMINAL command 371
PARTNER option
  CREATE PARTNER command 82
DISCARD PARTNER command 141
INQUIRE PARTNER command 280
PARTNER, CREATE command 81
PARTNER, DISCARD command 141
PARTNER, INQUIRE command 280
PATH option
  INQUIRE URIMAP command 429
PDSMEMBER option
  INQUIRE DOCTEMPLATE command 220
PENDSTATUS option
  INQUIRE CONNECTION command 190
  SET CONNECTION command 482
PERCLASS option
  INQUIRE MONITOR command 274
  SET MONITOR command 555
PERFORM CLASSCACHE command 443
  conditions 445
PERFORM commands
  CLASSCACHE 443
  DELETESHIPPED 450
  DUMP 453
  ENDAFFINITY 456
  RESETTIME 460
  SECURITY REBUILD 461
  SHUTDOWN 463
  STATISTICS RECORD 466
PERFORM CORBASERVER 447
PERFORM CORBASERVER command
  conditions 448
PERFORM DELETESHIPPED command 450
PERFORM DJAR 451
PERFORM DJAR command
  conditions 451
PERFORM DUMP command 453
  conditions 454
PERFORM ENDAFFINITY command 456
  conditions 456
PERFORM PIPELINE 458
PERFORM PIPELINE command
  conditions 458
PERFORM RESETTIME command 460
  conditions 460
PERFORM SECURITY REBUILD command 461
  conditions 461
PERFORM SHUTDOWN command 463
  conditions 464
PERFORM STATISTICS RECORD command 466
  conditions 469
PGMINTERFACE option
  INQUIRE WEBSERVICE command 438
PHASINGOUT option
  INQUIRE CLASSCACHE command 181
  INQUIRE JVMPOOL command 265
PHASINGOUTST option
  INQUIRE JVM command 263
PIPELINE CVDA value
  EXTRACT STATISTICS command 168
PIPELINE option
  CREATE PIPELINE command 84
  INQUIRE URIMAP command 429
  INQUIRE WEBSERVICE command 438
  PERFORM PIPELINE command 458
  PERFORM STATISTICS command 468
PIPELINE, CREATE command 83
PIPELINE, DISCARD command 142
PIPELINE, INQUIRE command 282
PIPELINE, PERFORM 458
PIPELINE, SET command 558
PL/I language
  argument values 12
  LENGTH option default 11
PLT option
  PERFORM SHUTDOWN command 464
  pointer arguments 8
  pointer-ref argument, CICS command format 6
  pointer-value argument, CICS command format 6
POOL option
  COLLECT STATISTICS command 44
POOL TARGET option
  COLLECT STATISTICS command 44
POOLNAME option
  INQUIRE TSQNAME command 405
  INQUIRE TSQUEUE command 405
PORT option
  INQUIRE TCPIPSERVICE command 347
PRINTADAPTST option
  INQUIRE TERMINAL command 371
PRINTCONTROL option
  INQUIRE TDQUEUE command 355
PRINTER option
  INQUIRE TDQUEUE command 355
PRIORITY option
  INQUIRE TASK command 333
  SET TERMINAL command 596
PRIVATE option
  INQUIRE TCPIPSERVICE command 347
PROCESS option
  INQUIRE TASK command 333
PROCESSTYPE option
  CREATE PROCESSTYPE command 86
  INQUIRE PROCESSTYPE command 285, 560
  INQUIRE TASK command 333
PROCESSTYPE, CREATE command 85
PROCESSTYPE, DISCARD command 143
PROCESSTYPE, INQUIRE command 284
PROCESSTYPE, SET command 559
PROFILE option
  CREATE PROFILE command 88
<table>
<thead>
<tr>
<th>PROFILE option (continued)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISCARD PROFILE command 144</td>
</tr>
<tr>
<td>INQUIRE CLASSCACHE command 181</td>
</tr>
<tr>
<td>INQUIRE JVM command 263</td>
</tr>
<tr>
<td>INQUIRE PARTNER command 280</td>
</tr>
<tr>
<td>INQUIRE PROFILE command 286</td>
</tr>
<tr>
<td>INQUIRE TASK command 333</td>
</tr>
<tr>
<td>INQUIRE TRANSACTION command 394</td>
</tr>
<tr>
<td>PERFORM CLASSCACHE command 444</td>
</tr>
<tr>
<td>PROFILE, DISCARD command 144</td>
</tr>
<tr>
<td>PROFILE, INQUIRE command 286</td>
</tr>
<tr>
<td>PROGAUTO option</td>
</tr>
<tr>
<td>COLLECT STATISTICS command 44</td>
</tr>
<tr>
<td>PERFORM STATISTICS command 468</td>
</tr>
<tr>
<td>PROGAUTOCTLG option</td>
</tr>
<tr>
<td>INQUIRE SYSTEM command 324</td>
</tr>
<tr>
<td>SET SYSTEM command 577</td>
</tr>
<tr>
<td>PROGAUTOEXIT option</td>
</tr>
<tr>
<td>INQUIRE SYSTEM command 324</td>
</tr>
<tr>
<td>SET SYSTEM command 577</td>
</tr>
<tr>
<td>PROGAUTOINST option</td>
</tr>
<tr>
<td>INQUIRE SYSTEM command 324</td>
</tr>
<tr>
<td>SET SYSTEM command 578</td>
</tr>
<tr>
<td>PROGRAM option</td>
</tr>
<tr>
<td>COLLECT STATISTICS command 44</td>
</tr>
<tr>
<td>CREATE PROGRAM command 90</td>
</tr>
<tr>
<td>DISABLE PROGRAM command 124</td>
</tr>
<tr>
<td>DISCARD PROGRAM command 145</td>
</tr>
<tr>
<td>ENABLE PROGRAM command 160</td>
</tr>
<tr>
<td>EXTRACT EXIT command 165</td>
</tr>
<tr>
<td>INQUIRE AUTOINSTALL command 172</td>
</tr>
<tr>
<td>INQUIRE DOCTEMPLATE command 220</td>
</tr>
<tr>
<td>INQUIRE PROGRAM command 293</td>
</tr>
<tr>
<td>INQUIRE TASK command 333</td>
</tr>
<tr>
<td>INQUIRE TRANSACTION command 394</td>
</tr>
<tr>
<td>INQUIRE URIMAP command 429</td>
</tr>
<tr>
<td>INQUIRE WEBSERVICE command 438</td>
</tr>
<tr>
<td>PERFORM STATISTICS command 468</td>
</tr>
<tr>
<td>SET AUTOINSTALL command 475</td>
</tr>
<tr>
<td>SET PROGRAM command 563</td>
</tr>
<tr>
<td>PROGRAM, CREATE command 89</td>
</tr>
<tr>
<td>PROGRAM, DISABLE command 123</td>
</tr>
<tr>
<td>PROGRAM, DISCARD command 145</td>
</tr>
<tr>
<td>PROGRAM, ENABLE command 157</td>
</tr>
<tr>
<td>PROGRAM, INQUIRE command 287</td>
</tr>
<tr>
<td>PROGRAM, SET command 561</td>
</tr>
<tr>
<td>PROGSYMBOLST option</td>
</tr>
<tr>
<td>INQUIRE TERMINAL command 371</td>
</tr>
<tr>
<td>QUALIFIER option</td>
</tr>
<tr>
<td>INQUIRE EXITPROGRAM command 239</td>
</tr>
<tr>
<td>QUALITY option</td>
</tr>
<tr>
<td>INQUIRE UOWENQ command 420</td>
</tr>
<tr>
<td>QUERYSYS option</td>
</tr>
<tr>
<td>INQUIRE TERMINAL command 371</td>
</tr>
<tr>
<td>QUEUE option</td>
</tr>
<tr>
<td>INQUIRE REQID command 298</td>
</tr>
<tr>
<td>QUEUEED option</td>
</tr>
<tr>
<td>INQUIRE TRANCLASS command 386</td>
</tr>
<tr>
<td>QUIESCESTATE option</td>
</tr>
<tr>
<td>INQUIRE DSNAME 225</td>
</tr>
</tbody>
</table>

Q

QALL option
- ACQUIRE TERMINAL command 36
- QNOTETAB option
- ACQUIRE TERMINAL command 36
- QSESSLIM option
- ACQUIRE TERMINAL command 36

QUALIFIER option
- INQUIRE EXITPROGRAM command 240
- RESYNC command 473

QUALLEN option
- INQUIRE UOWENQ command 420
- QUERY SECURITY command 18

R

RACF (resource access control facility) 2

READ option
- INQUIRE FILE command 249
READ option (continued)
   SET FILE command 536
READINTEG option
   INQUIRE FILE command 249
   SET FILE command 536
RECEIVECOUNT option
   INQUIRE CONNECTION command 191
RECFORMAT option
   INQUIRE FILE command 250
   INQUIRE TDQUEUE command 355
RECORDING option
   INQUIRE STATISTICS command 306
   SET STATISTICS command 568
RECORDLENGTH option
   INQUIRE TDQUEUE command 355
   RECORDNOW option
   SET STATISTICS command 568
RECORDSIZE option
   INQUIRE FILE command 250
   SET FILE command 537
RECOVERY option
   PERFORM STATISTICS command 468
RECOVERSTATUS option
   INQUIRE CONNECTION command 191
   INQUIRE DSNAME command 226
   INQUIRE FILE command 250
   INQUIRE TDQUEUE command 356
   INQUIRE TSNAME command 405
   INQUIRE TSQUEUE command 405
   SET CONNECTION command 484
REDIRECTTYPE option
   INQUIRE URIMAP command 430
   SET URIMAP command 627
REENTPROTECT option
   INQUIRE SYSTEM command 324
REGERROR value 346
REGISTERED value 346
RELATION option
   INQUIRE UOWENQ command 420
RELEASE option
   INQUIRE SYSTEM command 325
RELOAD option
   PERFORM CLASSCACHE command 444
   relocatable expression 13
RELREQ option
   ACQUIRE TERMINAL command 36
RELEQUEST option
   INQUIRE TERMINAL command 372
   SET TERMINAL command 597
RELTYPE option
   INQUIRE FILE command 250
   remote definition, not retrievable or updateable 1
REMOTENAME option
   INQUIRE CONNECTION command 192
   INQUIRE FILE command 251
   INQUIRE PROGRAM command 293
   INQUIRE TASK command 333
   INQUIRE TDQUEUE command 356
   INQUIRE TERMINAL command 372
   INQUIRE TRANSACTION command 394
REOTESYSNET option
   INQUIRE BRFACILITY command 176
   INQUIRE CONNECTION command 192
   INQUIRE TERMINAL command 372
REOTESYSTEM option
   INQUIRE BRFACILITY command 176
   INQUIRE CONNECTION command 192
   INQUIRE FILE command 251
   INQUIRE PROGRAM command 293
   INQUIRE TASK command 334
   INQUIRE TDQUEUE command 356
   INQUIRE TERMINAL command 372
   INQUIRE TRANSACTION command 395
REMOTETABLE option
   INQUIRE FILE command 251
   removing 517
REQID option
   INQUIRE REQID command 298
   REQID, INQUIRE command 296
REQTYPE option
   INQUIRE REQID command 298
REQUESTID option
   INQUIRE WORKREQUEST command 441
REQUESTMODEL option
   CREATE REQUESTMODEL command 93
   PERFORM STATISTICS command 468
REQUESTMODEL, CREATE command 92
REQUESTMODEL, DISCARD command 146
REQUESTMODEL, INQUIRE command 301
REQUIRED value
   INQUIRE TCPIPSERVICE command 347
RESCOUNT option
   INQUIRE PROGRAM command 293
RESETNOW option
   PERFORM STATISTICS command 468
   SET STATISTICS command 568
   RESETTIME, PERFORM command 460
RESLEN option
   INQUIRE UOWENQ command 420
RESNAME option
   INQUIRE TASK command 334
   resource access control facility (RACF) 2
   RESOURCE option
   INQUIRE UOWENQ command 420
   resource security checking 17
   resources
      class (ESM) 16
      RESP and RESP2 options
      values returned 16
      RESP options 15
      RESP2 option 15
      RESP2 options 15
      RESP2 values
         EXEC CICS CREATE 681
Response Codes
   of EXEC CICS commands 671
RESRCCLASS option
   INQUIRE MONITOR command 274
   SET MONITOR command 555
RESSEC option
  INQUIRE TASK command 334
  INQUIRE TRANSACTION command 395
RESTYPE option
  EXTRACT STATISTICS command 168
RESYNC command
  ENTRYNAME 472
  IDLIST 472
  IDLISTLENGTH 472
  PARTIAL 472
  QUALIFIER 473
RESYNC ENTRYNAME command 471
  conditions 473
RESYNCSTATUS option
  INQUIRE UOWLINK command 424
RELOCKS option
  INQUIRE DSNAME 226
REUSEST option
  INQUIRE CLASSCACHE command 181
  INQUIRE JVM command 263, 268
REWIND option
  INQUIRE TDQUEUE command 356
RLSACCESS option
  INQUIRE FILE command 251
  SET FILE command 537
RMQFY option
  INQUIRE UOWLINK command 425
RMIST option
  INQUIRE MONITOR command 274
ROLE option
  INQUIRE UOWLINK command 425
ROUTESTATUS option
  INQUIRE TRANSACTION command 395
ROUTING option
  INQUIRE TASK command 334
  INQUIRE TRANSACTION command 395
RRMS, INQUIRE command 303
RTERMID option
  INQUIRE REQID command 298
RTIMEOUT option
  INQUIRE TASK command 334
  INQUIRE TRANSACTION command 395
TRANSID option
  INQUIRE REQID command 298
rules for browsing 23
RUNAWAY option
  INQUIRE DISPATCHER command 216
  INQUIRE SYSTEM command 325
  INQUIRE TASK command 335
  INQUIRE TRANSACTION command 395
  SET DISPATCHER command 514
  SET SYSTEM command 578
  SET TRANSACTION command 618
RUNAWAYTYPE option
  INQUIRE TRANSACTION command 396
  SET TRANSACTION command 618
RUNNING option
  INQUIRE TASK LIST command 339
RUNSTATUS option
  INQUIRE TASK command 335
RUNTIME option
  INQUIRE PROGRAM command 293
  SET PROGRAM command 563
S
SCANDELEY option
  INQUIRE DISPATCHER command 216
  INQUIRE SYSTEM command 325
  SET DISPATCHER command 514
  SET SYSTEM command 578
SCHEME option
  INQUIRE UURIMAP command 429
SCRNHT option
  INQUIRE TERMINAL command 372
SCRNSIZE option
  INQUIRE TASK command 335
SCRNWDO option
  INQUIRE TERMINAL command 372
SDASIZE option
  INQUIRE SYSTEM command 325
SDTRAN option
  INQUIRE SYSTEM command 325
  PERFORM SHUTDOWN command 464
SECONDS option
  INQUIRE REQID command 299
security 19
  command 16
  NOTAUTH condition 16
  QUERY SECURITY command 18
  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 372
SECURITY REBUILD, PERFORM command 461
SECURITYMGR option
  INQUIRE SYSTEM command 325
SENDCOUNT option
  INQUIRE CONNECTION command 192
SERVICE option
  INQUIRE TCPIPSERVICE command 347
SERVSTATUS option
  INQUIRE CONNECTION command 192
  INQUIRE TERMINAL command 372
  SET CONNECTION command 484
  SET TERMINAL command 598
SESSBEANTIME option
  INQUIRE CORBASERVER command 196
  SET CORBASERVER command 490
SESSIONS option
  CREATE SESSIONS command 95
SESSIONTYPE option
  INQUIRE TERMINAL command 373
  SET AUTOINSTALL command 474
SET AUTOINSTALL command
(continued)
  conditions 475
SET BRFACILITY command 476
  conditions 476
SET CLASSCACHE command 478
  conditions 479
SET commands 23
  AUTOINSTALL 474
  BRFACILITY 476
  CLASSCACHE 478
  CONNECTION 480
  CORBASERVER 489
  DB2CONN 492
  DB2ENTRY 502
  DB2TRAN 508
  DELETSHPIPPED 510
  DISPATCHER 513
  DSNAME 516
  DUMPDS 525
  FILE 529
  IRC 544
  JOURNALNUM 548
  MODENAME 551
  MONITOR 553
  NETNAME 557
  PIPELINE 558
  PROGRAM 561
  STATISTICS 566
  SYSDUMPCODE 570
  SYSTEM 574
  TASK 580
  TCLASS 582
  TCPIP 583
  TCPIPSERVICE 585
  TDQUEUE 588
  TERMINAL 593
  TRACEDEST 602
  TRACEFLAG 606
  TRACETYPE 608
  TRANCLASS 611
  TRANDUMPCODE 613
  TRANSACTION 617
  UOWLINK 625
  VOLUME 629
  VTAM 630
SET CONNECTION command 480
  conditions 486
SET CORBASERVER command 489
  conditions 490
SET DB2CONN command 492
  conditions 499
SET DB2ENTRY command 502
  conditions 506
SET DB2TRAN command 508
  conditions 508
SET DELETSHPIPPED command 510
  conditions 511
SET DISPATCHER command 513
  conditions 515
SET DSNAME command 516
  conditions 521
SET DUMPDS command 525
  conditions 526
SET ENQMODEL command 527
  conditions 527
SET FILE command 529
  conditions 538
SET HOST command 542
  conditions 543
SET IRC command 544
  conditions 544
SET JOURNALNAME command 546
  conditions 547
SET JOURNALNUM command 548
SET JVMPOOL command 549
  conditions 550
SET MODENAME command 551
  conditions 552
SET MONITOR command 553
  conditions 556
SET NETNAME command 557
  conditions 557
SET option
  COLLECT STATISTICS command 44
  EXTRACT STATISTICS command 168
  INQUIRE REQID command 299
  INQUIRE TASK LIST command 339
SET PIPELINE command 558
  conditions 558
SET PROCESSTYPE command 559
SET PROGRAM command 561
  conditions 564
SET STATISTICS command 566
  conditions 568
SET SYSNAME command 570
  conditions 572
SET SYSTEM command 574
  conditions 578
SET TASK command 580
  conditions 581
SET TCLASS command 582
  conditions 582
SET TCPIP command 583
  conditions 584
SET TCPIPSERVICE command 585
  conditions 586
SET TDQUEUE command 588
  conditions 591
SET TERMINAL command 593
  conditions 600
SET TRACEDEST command 602
  conditions 604
SET TRACEFLAG command 606
  conditions 607
SET TRACETYPE command 608
  conditions 610
SET TRANCLASS command 611
  conditions 612
SET TRANDUMPCODE command 613
  conditions 615
SET TRANSACTION command 617
  conditions 619
SET TSQNAME command 621
   conditions 621
SET TSQUEUE command 621
   conditions 621
SET UOW command 623
   conditions 623
SET UOWLINK command 625
   conditions 625
SET URIMAP command 627
   conditions 628
SET VOLUME command 629
   conditions 629
SET VTAM command 630
   conditions 632
SET WEBSERVICE command 634
   conditions 634
SET WORKREQUEST command 635
   conditions 636
SETTRANSID option
   INQUIRE TASK LIST command 339
SHARESTATUS option
   INQUIRE PROGRAM command 294
   SET PROGRAM command 563
SHELF option
   INQUIRE CORBASERVER command 196
SHUTDOWN option
   DISABLE PROGRAM command 124
   ENABLE PROGRAM command 160
   INQUIRE TRANSACTION command 396
   SET TRANSACTION command 618
SHUTDOWN, PERFORM command 463
SHUTDOWNST option
   INQUIRE EXITPROGRAM command 240
SHUTOPTION option
   INQUIRE SYSDUMPCODE command 315
   INQUIRE TRANDUMPCODE command 389
   SET SYSDUMPCODE command 571
   SET TRANDUMPCODE command 614
SHUTSTATUS option
   INQUIRE SYSTEM command 326
SIGNID option of DSNCRCT macro 200, 206
SIGNONSTATUS option
   INQUIRE TERMINAL command 373
SINGLESTATUS option
   INQUIRE TRACEFLAG command 381
   SET TRACEFLAG command 606
SOCKETCLOSE option
   INQUIRE TCPIPSERVICE command 348
SOSIST option
   INQUIRE TERMINAL command 373
SOSTATUS option
   INQUIRE SYSTEM command 326
SPECIAL option
   INQUIRE TRACETYPE command 385
   SET TRACETYPE command 610
SPI option
   DISABLE PROGRAM command 124
   ENABLE PROGRAM command 160
SPIST option
   INQUIRE EXITPROGRAM command 240
SSL option
   PERFORM CORBASERVER command 447
   PERFORM DJAR command 451
SSL value
   INQUIRE TCPIPSERVICE command 348
   SSLCACHE option
   INQUIRE TCPIP command 342
   SSLTYPE option
   INQUIRE TCPIPSERVICE command 348
   SSLUNAUTH option
   INQUIRE CORBASERVER command 196
STACK option
   INQUIRE WORKREQUEST command 441
   STANDARD option
   INQUIRE JVMPOOL command 265
   INQUIRE PROCESS command 285, 560
   INQUIRE PROGRAM command 294
   INQUIRE TCPIPSERVICE command 347
   INQUIRE TRANSACTION command 396
   SET JVMPOOL command 549
   SET MONITOR command 556
   SET PROGRAM command 563
   SET TRANSACTION command 619
START option
   ENABLE PROGRAM command 160
   PERFORM CLASSCACHE command 444
STARTCODE option
   INQUIRE TASK command 335
   starting a browse 21
STARTSTATUS option
   INQUIRE EXITPROGRAM command 240
STARTTIME option
   INQUIRE CLASSCACHE command 181
STARTUP option
   INQUIRE SYSTEM command 326
STARTUPDATE option
   INQUIRE SYSTEM command 326
STATE option
   INQUIRE DJAR command 218
   INQUIRE UOWENQ command 420
   INQUIRE WEBSERVICE command 438
STATISTICS RECORD, PERFORM command 466
STATISTICS, COLLECT command 38
STATISTICS, EXTRACT command 166
STATISTICS, INQUIRE command 304
STATISTICS, SET command 566
STATS option
   COLLECT STATISTICS command 44
   PERFORM STATISTICS command 468
STATUS option
   INQUIRE CLASSCACHE command 181
   INQUIRE ENQMODEL command 232
   INQUIRE JVMPOOL command 265
   INQUIRE MONITOR command 274
   INQUIRE PROCESS command 285, 560
   INQUIRE PROGRAM command 294
   INQUIRE TCPIPSERVICE command 347
   INQUIRE TRANSACTION command 396
   SET JVMPOOL command 549
   SET MONITOR command 556
   SET PROGRAM command 563
   SET TRANSACTION command 619
STOP option
   DISABLE PROGRAM command 124
STORAGE option
   COLLECT STATISTICS command 44
   PERFORM STATISTICS command 468
   STORAGE, INQUIRE command 307

Index 727
STORAGECLEAR option
  INQUIRE TASK command 336
  INQUIRE TRANSACTION command 396
STOREPROTECT option
  INQUIRE SYSTEM command 327
STREAMNAME option
  PERFORM STATISTICS command 468
STREAMNAME, INQUIRE command 310
STRINGS option
  INQUIRE FILE command 251
  SET FILE command 537
SUBPOOL option
  COLLECT STATISTICS command 44
  INQUIRE SUBPOOL command 312
SUBPOOL, INQUIRE command 312
SUBPOOLLIST option
  INQUIRE MVSTCB command 277
SUBSYSTEMID option
  INQUIRE MONITOR command 274
SUBTASKS option
  INQUIRE DISPATCHER command 216
SUPPORTED value
  INQUIRE TCPIPSERVICE command 347
surrogate security checking 16
SUSPENDED option
  INQUIRE TASK LIST command 340
SUSPENDTIME option
  INQUIRE TASK command 336
SUSPENDTYPE option
  INQUIRE TASK command 336
SUSPENDVALUE option
  INQUIRE TASK command 336
SWITCHACTION option
  SET TRACEDEST command 603
SWITCHSTATUS option
  INQUIRE DUMPDS command 229
  INQUIRE TRACEDEST command 379
  SET DUMPDS command 526
  SET TRACEDEST command 603
SYNCPOINTST option
  INQUIRE MONITOR command 274
  SET MONITOR command 556
syntax notation 3
SYSDUMP option
  PERFORM STATISTICS command 468
SYSDUMPCODE option
  COLLECT STATISTICS command 45
  SET SYSDUMPCODE command 572
SYSDUMPCODE, INQUIRE command 314
SYSDUMPCODE, SET command 570
SYSDUMPING option
  INQUIRE SYSDUMPCODE command 315
  INQUIRE TRANDUMPCODE command 389
  SET SYSDUMPCODE command 572
  SET TRANDUMPCODE command 614
SYSEIB option
  3
SYSID option
  INQUIRE TSQNAME command 405
  INQUIRE TSQUEUE command 405
  INQUIRE UOW command 409
SYSID option (continued)
  INQUIRE UOWLINK command 425
SYOUTCLASS option
  INQUIRE TDQUEUE command 356
SYSPLEX value
  INQUIRE TCPIP command 343
system connections 128, 142, 150, 156, 183
system programming commands 1
  inquiry 19
SYSTEM, INQUIRE command 317
SYSTEM, SET command 574
systemname argument, CICS command format 6
SYSTEMSTATUS option
  INQUIRE TRACEFLAG command 381
  SET TRACEFLAG command 606
T
TABLE option
  INQUIRE FILE command 251
  SET FILE command 537
TABLEMGR option
  COLLECT STATISTICS command 45
  PERFORM STATISTICS command 469
TABLENAME option
  INQUIRE FILE command 251
  SET FILE command 538
TABLESIZE option
  INQUIRE TRACEDEST command 380
  SET TRACEDEST command 604
TAKEOVER option
  PERFORM SHUTDOWN command 464
TALENGTH option
  ENABLE PROGRAM command 161
  INQUIRE EXITPROGRAM command 240
TARGET NODE option
  COLLECT STATISTICS command 45
TASK LIST, INQUIRE command 339
TASK option
  INQUIRE EXCI command 234
  INQUIRE JVM command 264
  INQUIRE STORAGE command 308
  INQUIRE UOW command 409
  INQUIRE WORKREQUEST command 441
  SET TASK command 581
task-related user exits, restart resynchronization 471
TASK, INQUIRE command 328
TASK, SET command 580
TASKDATAKEY option
  INQUIRE TASK command 336
  INQUIRE TRANSACTION command 396
TASKDATALOC option
  INQUIRE TASK command 337
  INQUIRE TRANSACTION command 397
TASKID option
  INQUIRE BRFACILITY command 176
  INQUIRE TERMINAL command 373
  INQUIRE UOWENQ command 420
TASKSTART option
  DISABLE PROGRAM command 125
  ENABLE PROGRAM command 161
TASKSTARTST option
  INQUIRE EXITPROGRAM command 241
TASKSUBPOOL option
  COLLECT STATISTICS command 45
TCAMCONTROL option
  INQUIRE TERMINAL command 373
  SET TERMINAL command 598
TCB option
  INQUIRE TASK command 337
TEXITSTATUS option
  INQUIRE TRACEFLAG command 382
  SET TRACEFLAG command 607
TCLASS option
  COLLECT STATISTICS command 45
  INQUIRE TASK command 337
  INQUIRE TRANSACTION command 397
  PERFORM STATISTICS command 469
  SET TRANSACTION command 619
TCLASS, INQUIRE command 341
TCLASS, SET command 582
TCPIP option
  COLLECT STATISTICS command 45
  PERFORM STATISTICS command 469
TCPIP, INQUIRE command 342
TCPIP, SET command 583
TCPIPSERVICE option
  COLLECT STATISTICS command 45
  CREATE TCPIPSERVICE command 98
  INQUIRE TCPIPSERVICE command 348
  INQUIRE URIMAP command 255, 429
  PERFORM STATISTICS command 469
TCPIPSERVICE, DISCARD command 147
TCPIPSERVICE, INQUIRE command 344
TCPIPSERVICE, SET command 585
TDQ option
  INQUIRE DOCTEMPLATE command 220
TDQUEUE option
  COLLECT STATISTICS command 45
  CREATE TDQUEUE command 102
  DISCARD TDQUEUE command 148
  INQUIRE DOCTEMPLATE command 220
  INQUIRE TDQUEUE command 356
  PERFORM STATISTICS command 469
  SET TDQUEUE command 590
TDQUEUE, DISCARD command 148
TDQUEUE, INQUIRE command 350
TDQUEUE, SET command 588
TEMPLATENAME option
  INQUIRE DOCTEMPLATE command 220
  INQUIRE URIMAP command 429
TEMPLATETYPE option
  INQUIRE DOCTEMPLATE command 220
TERM option
  DSNCRCT macro 200, 206, 494, 504
TERMINAL option (continued)
  CREATE TERMINAL command 106
  INQUIRE TERMINAL command 374
  PERFORM STATISTICS command 469
  SET TERMINAL command 598
  TERMINAL, ACQUIRE command 35
  TERMINAL, CREATE command 104
  TERMINAL, DISCARD command 150
  TERMINAL, INQUIRE command 358
  TERMINAL, SET command 593
TERMINATE option
  PERFORM CLASSCACHE command 445
TERMINATION option
  SET JVMPOOL command 549
TERMMODEL option
  INQUIRE TERMINAL command 374
TERMPRIORITY option
  INQUIRE TERMINAL command 374
  SET TERMINAL command 598
TERMSTATUS option
  INQUIRE BRFACILITY command 176
  INQUIRE TERMINAL command 374
  SET BRFACILITY command 476
  SET TERMINAL command 598
TEXTKYBDST option
  INQUIRE TERMINAL command 374
TEXTPRINTFST option
  INQUIRE TERMINAL command 374
Threadsafenew
  EXEC CICS SPI commands 697
TIME option
  INQUIRE DISPATCHER command 216
  INQUIRE MONITOR command 275
  INQUIRE REQID command 299
  INQUIRE SYSTEM command 327
  SET DISPATCHER command 514
  SET SYSTEM command 578
  timeout delete mechanism 212
TIMEOUT option
  INQUIRE TCPIPSERVICE command 348
TIMEOUTINT option
  CEMT INQUIRE WEB 436
  SET WEB 633
TITLE option
  PERFORM DUMP command 454
TITLELENGTH option
  PERFORM DUMP command 454
TOTAL option
  INQUIRE JVMPOOL command 266
TOTALJVMS option
  INQUIRE CLASSCACHE command 182
TPNAME option
  INQUIRE PARTNER command 280
TPNAMELEN option
  INQUIRE PARTNER command 280
TRACEDEST, INQUIRE command 378
TRACEDEST, SET command 602
TRACEFLAG, INQUIRE command 381
TRACEFLAG, SET command 606
TRACETYPE, INQUIRE command 383
TRACETYPE, SET command 608
TRACING option
INQUIRE TASK command 337
INQUIRE TERMINAL command 374
INQUIRE TRANSACTION command 397
SET TERMINAL command 599
SET TRANSACTION command 619
TRANCLASS option
COLLECT STATISTICS command 45
CREATE TRANCLASS command 108
DISCARD TRANCLASS command 152
INQUIRE TASK command 338
INQUIRE TRANCLASS command 386
INQUIRE TRANSACTION command 397
PERFORM STATISTICS command 469
SET TRANCLASS command 612
SET TRANSACTION command 619
TRANCLASS, CREATE command 107
TRANCLASS, DISCARD command 152
TRANCLASS, INQUIRE command 386
TRANCLASS, SET command 611
TRANDUMP
PERFORM STATISTICS command 469
TRANDUMPCODE option
COLLECT STATISTICS command 45
INQUIRE TRANDUMPCODE command 390
SET TRANDUMPCODE command 615
TRANDUMPCODE, INQUIRE command 388
TRANDUMPCODE, SET command 613
TRANDUMPING option
INQUIRE TRANDUMPCODE command 390
SET TRANDUMPCODE command 615
TRANISOLATE option
INQUIRE SYSTEM command 327
TRANPRIORITY option
INQUIRE TASK command 338
TRANSACTION option
COLLECT STATISTICS command 45
CREATE TRANSACTION command 110
DISCARD TRANSACTION command 153
INQUIRE BRFACILITY command 177
INQUIRE TASK command 338
INQUIRE TERMINAL command 375
INQUIRE TRANSACTION command 397
INQUIRE UIRMAP command 429
PERFORM STATISTICS command 469
SET TRANSACTION command 619
transaction security checking 16
TRANSACTION, DISCARD command 153
TRANSACTION, INQUIRE command 391
TRANSACTION, SET command 617
TRANSID option
INQUIRE PROGRAM command 294
INQUIRE REQID command 299
INQUIRE REQUESTMODEL command 302
INQUIRE TCPIPSERVICE command 348
INQUIRE TSQUEUE command 405
INQUIRE TSQUEUE command 405
INQUIRE UOW command 409
INQUIRE UOWENQ command 420
INQUIRE WORKREQUEST command 441
translator 1
TRIGGERLEVEL option
INQUIRE TDQUEUE command 356
SET TDQUEUE command 590
TRPROF option
INQUIRE TASK command 338
INQUIRE TRANSACTION command 398
TSMODEL option
CREATE TSMODEL command 113
DISCARD TSMODEL command 154
TSMODEL, DISCARD command 154
TSMODEL, INQUIRE command 399
TSPool option
INQUIRE TSPool command 401
TSPool, INQUIRE command 401
TS option
INQUIRE DOCTEMPLATE command 220
TSQNAME
INQUIRE TSQNAME command 405
TSQNAME, INQUIRE command 403
TSQPREFIX option
INQUIRE TCPIPSERVICE command 348
TSQUEUE
INQUIRE TSQUEUE command 405
TSQUEUE option
COLLECT STATISTICS command 45
INQUIRE DOCTEMPLATE command 220
PERFORM STATISTICS command 469
TSQUEUE, INQUIRE command 403
TSQUEUELIMIT option
INQUIRE MONITOR command 275
TSYSTEM option
INQUIRE WORKREQUEST command 441
TTSTATUS option
INQUIRE TERMINAL command 375
SET TERMINAL command 599
TWAIT option of DSNCRCT macro
TYPE=ENTRY macro 499
TWASIZE option
INQUIRE TASK command 338
INQUIRE TRANSACTION command 398
TX
option of DSNCRCT macro 199
TX option of DSNCRCT macro 494
TXID option of DSNCRCT macro 200, 207, 504
TYPE option
INQUIRE DOCTEMPLATE command 221
INQUIRE FILE command 252
INQUIRE REQUESTMODEL command 302
INQUIRE TDQUEUE command 356
INQUIRE UOWENQ command 420
INQUIRE UOWLINK command 425
TYPETERM option
CREATE TYPETERM command 116
TYPETERM, CREATE command 114
U
UCTRANST option
INQUIRE TERMINAL command 375
SET TERMINAL command 599
INDEX

UDASIZE option
  INQUIRE SYSTEM command 327
UNAUTH option
  INQUIRE CORBASERVER command 196
UNAVAILABLE value 346
UNREGISTERED value 346
UOW option
  INQUIRE TASK command 338
  INQUIRE UOW command 409
  INQUIRE UOWENQ command 422
  INQUIRE UOWLINK command 425
UOW, INQUIRE command 407
UOWACTION option
  SET CONNECTION command 485
UOWDSNFAIL, INQUIRE command 412
UOWENQ, INQUIRE command 417
UOWLINK option
  INQUIRE UOWLINK command 425
  SET UOWLINK command 625
UOWLINK, INQUIRE command 423
UOWSTATE option
  INQUIRE UOW command 409
UPDATE option
  INQUIRE FILE command 252
  SET FILE command 538
UPDATEMODEL option
  INQUIRE FILE command 252
  SET FILE command 538
URID option
  INQUIRE EXCI command 234
  INQUIRE UOW command 425
URIMAP CVDA value
  EXTRACT STATISTICS command 168
URIMAP option
  CREATE URIMAP command 119
  DISCARD URIMAP command 155
  INQUIRE URIMAP command 427
  INQUIRE WEBSERVICE command 439
  PERFORM STATISTICS command 469
URIMAP, CREATE command 118
URM option
  INQUIRE TCPIPSERVICE command 348
  SET TCPIPSERVICE command 586
USAGE option
  INQUIRE URIMAP command 430
USECOUNT option
  INQUIRE EXITPROGRAM command 241
  INQUIRE PROGRAM command 294
USER
  option of DSNCRCT macro 207
USER value
  INQUIRE TCPIPSERVICE command 347
USERAREA option
  INQUIRE TERMINAL command 375
USERAREALEN option
  INQUIRE TERMINAL command 375
USERDATA option
  ACQUIRE TERMINAL command 36
USERDATALEN option
  ACQUIRE TERMINAL command 36
USERID
  option of DSNCRCT macro 199
USERID option
  INQUIRE BRFACILITY command 177
  INQUIRE REQID command 299
  INQUIRE TASK command 338
  INQUIRE TERMINAL command 376
  INQUIRE UURIMAP command 430
USERID option of DSNCRCT macro 207, 504
USERNAME option
  INQUIRE TERMINAL command 376
USERSTATUS option
  INQUIRE TERMINAL command 376
V
VALIDATIONST option
  INQUIRE TERMINAL command 376
  INQUIRE WEBSERVICE command 439, 634
VALIDITY option
  INQUIRE DSNAME command 227
VERSION option
  SET PROGRAM command 563
VFORMST option
  INQUIRE TERMINAL command 376
VOLUME, INQUIRE command 432
VOLUME, SET command 629
VTAM option
  COLLECT STATISTICS command 45
  PERFORM STATISTICS command 469
VTAM, INQUIRE command 433
VTAM, SET command 630
W
WAIT value
  INQUIRE TCPIPSERVICE command 348
WAITCAUSE option
  INQUIRE UOW command 409
WAITSTATE option
  INQUIRE UOW command 410
Web support
  INQUIRE transaction 436
  SET command 633
  WEB 633
WEBSERVICE CVDA value
  EXTRACT STATISTICS command 168
WEBSERVICE option
  CREATE WEBSERVICE command 122
  INQUIRE URIMAP command 430
  INQUIRE WEBSERVICE command 439, 634
  PERFORM STATISTICS command 469
  WEBSERVICE, CREATE command 121
  WEBSERVICE, DISCARD command 156
  where-clause, CICS command format 6
WORKREQUEST option
  INQUIRE WORKREQUEST command 442
WORKREQUEST, INQUIRE command 440
WORKTYPE option
  INQUIRE WORKREQUEST command 442
WORKTYPE option (continued)
  SET WORKREQUEST command 636
WSBIND option
  INQUIRE WEBSERVICE command 439
WSDLFILE option
  INQUIRE WEBSERVICE command 439

X
XLNSTATUS option
  INQUIRE CONNECTION command 192
XLT option
  PERFORM SHUTDOWN command 464
XRFSTATUS option
  INQUIRE SYSTEM command 327

Z
ZCPTREADING option
  INQUIRE CONNECTION command 193
  INQUIRE TERMINAL command 376
  SET CONNECTION command 486
  SET TERMINAL command 600
Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply in the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.
The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Programming interface information

This book is intended to help you use the CICS system programming commands. This book primarily documents General-use Programming Interface and Associated Guidance Information provided by CICS.

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

However, this book also documents Product-sensitive Programming Interface and Associated Guidance Information provided by CICS.

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

Product-sensitive Programming Interface and Associated Guidance Information is identified where it occurs by an introductory statement to a chapter or section.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Adobe and the Adobe logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.
Sending your comments to IBM

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book.

Please limit your comments to the information in this book and the way in which the information is presented.

To ask questions, make comments about the functions of IBM products or systems, or to request additional publications, contact your IBM representative or your IBM authorized remarketer.

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

You can send your comments to IBM in any of the following ways:
• By mail, to this address:
  IBM United Kingdom Limited
  User Technologies Department (MP095)
  Hursley Park
  Winchester
  Hampshire
  SO21 2JN
  United Kingdom
• By fax:
  – From outside the U.K., after your international access code use
    44–1962–816151
  – From within the U.K., use 01962–816151
• Electronically, use the appropriate network ID:
  – IBMLink: HURSLEY(IDRCF)
  – Internet: idrcf@hursley.ibm.com

Whichever you use, ensure that you include:
• The publication title and order number
• The topic to which your comment applies
• Your name and address/telephone number/fax number/network ID.