CICS Transaction Server for z/OS

CICS Supplied Transactions

Version 3 Release 2

SC34-6817-03

CICS Transaction Server for z/OS

CICS Supplied Transactions

Version 3 Release 2

SC34-6817-03

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 615.

This edition applies to Version 3 Release 2 of CICS Transaction Server for z/OS, program number 5655-M15, and to all subsequent versions, releases, and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 1982, 2011.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

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

Summary of changes xiii
Changes for CICS Transaction Server for z/OS, Version 3 Release 2 xiii
Changes for CICS Transaction Server for z/OS, Version 3 Release 1 xiii
Changes for CICS Transaction Server for z/OS, Version 2 Release 3 xiv
Changes for CICS Transaction Server for z/OS, Version 2 Release 2 xiv
Changes for CICS Transaction Server for z/OS, Version 2 Release 1 xv

Chapter 1. Using CICS supplied transactions 1
How to start or stop a transaction 3
Syntax notation and conventions used 3
Minimum abbreviation of keywords 4
Uppercase input to transactions 4
Terminal operation 5
$ (the dollar symbol) 5
Operator security 5
Terminal operator 6
Supervisory terminal operator 6
Master terminal operator 7
MRO and ISC support 8
BMS partitions 8
Using the system console 10
Rules for console entry 10
MODIFY command 11
REPLY command 12
Example of a conversation using CONVERSE 12
Example of a conversation using WRITE/READ 12
Using TSO consoles 13
Using the KILL option for purging transactions 13
Using CEMT to terminate a task 14
Using the console to terminate a task 14
Consequences of using KILL 15

Chapter 2. The application debugging profile manager 17
Using the application debugging profile manager Web interface 17
Configuring access to the application debugging profile manager Web interface 17
Using the application debugging profile manager 3270 interface 37

Chapter 3. CBAM - BTS browser 57

Chapter 4. CCRL - certificate revocation lists transaction 59

Chapter 5. CDBC - database control menu 61
Connecting CICS to DBCTL 62
Disconnecting CICS from DBCTL 62
Specifying a DBCTL system identifier (DBCTLID) 63
Using the HELP panel 63

© Copyright IBM Corp. 1982, 2011

Chapter 6. CDBI - database control inquiry 65
Using the CDBI HELP panel 65

Chapter 7. CDBM - database control interface 67
Using CDBM Syntax 67
Using DFHDBFK - The CDBM GROUP command file 69
Record layout in the CDBM GROUP command file. 70
The MAINTENANCE panel for DFHDBFK 70
Using the CDBM HELP panels 72

Chapter 8. CEBR - temporary storage browse 75
Using the CEBR HELP panel 77

Chapter 9. CEBT - master terminal (alternate CICS) 79
Minimum abbreviation of keywords 79
CEBT INQUIRE and SET commands. 79
CEBT INQUIRE AUXTRACE 79
CEBT INQUIRE DUMP 80
CEBT INQUIRE GTFTRACE 81
CEBT INQUIRE INTTRACE 82
CEBT INQUIRE SURVEILLANCE 82
CEBT INQUIRE TAKEOVER 83
CEBT SET AUXTRACE. 83
CEBT SET DUMP. 84
CEBT SET GTFTRACE. 85
CEBT SET INTTRACE 86
CEBT SET SURVEILLANCE 86
CEBT SET TAKEOVER. 87
CEBT PERFORM commands 88
CEBT PERFORM SHUTDOWN 88
CEBT PERFORM SNAP 88
CEBT PERFORM TAKEOVER 88

Chapter 10. CECI - command-level interpreter 91
Using the CECI HELP panel 92
Command-level syntax-checking transaction (CECS) 93

Chapter 11. CEDA - resource definition online 95
Command syntax 95
CSD compatibility panel 101
Security checking 103
CEDA HELP panels 103
CEDB and CEDC 107

Chapter 12. CEDF and CEDX - the execution diagnostic facility 109
Using the CEDF transaction 109
Command syntax 109
Command options 109
Beginning an EDF session (single-screen mode) 110
Beginning an EDF session (dual-screen mode). 111
Monitoring remote transactions. 111
Invoking CEBR and CECI from CEDF 111
Using the CEDX transaction 112
Command syntax 113
Command options 113

Chapter 13. CEKL—master terminal (emergency use) 115
Minimum abbreviation of keywords 115
CEKL INQUIRE TASK 115
CEKL SET TASK 118

Chapter 14. CEMN - CICS monitoring facility 121

Chapter 15. CEMT—master terminal 123
Starting the CEMT transaction 123
System parameters 126
Program function (PF) keys 126
Minimum abbreviation of keywords 127
ALL option 127
CLASS option 127
Family of resources (using * and + symbols) 127
Subdefining resource groups 128
Overtyping a display 128
Tab key 129
Scrolling symbol (+ sign) 129
Blank fields in a display 129
Using the ? key 129
Expanded panels 130
Hexadecimal values 131
Variables. 131
Invoking CEMT from an application program 132
Security checking 132
CEMT DISCARD commands 132
CEMT DISCARD. 134
CEMT INQUIRE commands 138
CEMT INQUIRE AUTINSTMODEL 139
CEMT INQUIRE AUTOINSTALL 140
CEMT INQUIRE AUXTRACE 142
CEMT INQUIRE BEAN 144
CEMT INQUIRE BRFACILITY 146
CEMT INQUIRE CFDTPOOL 150
CEMT INQUIRE CLASSCACHE 151
CEMT INQUIRE CONNECTION 154
CEMT INQUIRE CORBASERVER 163
CEMT INQUIRE DB2CONN 167
CEMT INQUIRE DB2ENTRY 174
CEMT INQUIRE DB2TRAN 178
CEMT INQUIRE DELETSHIPPED 179
CEMT INQUIRE DISPATCHER 181
CEMT INQUIRE DJAR 183
CEMT INQUIRE DOCTEMPLATE 186
CEMT INQUIRE DSAS 188
CEMT INQUIRE DSNAME 191
CEMT INQUIRE DUMPDS 199
CEMT INQUIRE ENQ 200
CEMT INQUIRE ENQMODEL 204
CEMT INQUIRE EXCI 206
CEMT INQUIRE FEPI 208
CEMT INQUIRE FILE 208
CEMT INQUIRE GTFTRACE 215
CEMT INQUIRE HOST 216
CEMT INQUIRE INTTRACE 218

CEMT INQUIRE IPCONN 219
CEMT INQUIRE IRBATCH 223
CEMT INQUIRE IRC 224
CEMT INQUIRE JMODEL 225
CEMT INQUIRE JOURNALNAME 227
CEMT INQUIRE JVM 229
CEMT INQUIRE JVMPOOL. 231
CEMT INQUIRE LIBRARY 232
CEMT INQUIRE LINE 235
CEMT INQUIRE MODENAME 236
CEMT INQUIRE MONITOR 238
CEMT INQUIRE NETNAME 241
CEMT INQUIRE PARTNER 247
CEMT INQUIRE PIPELINE 249
CEMT INQUIRE PROCESSTYPE 251
CEMT INQUIRE PROFILE 253
CEMT INQUIRE PROGRAM 254
CEMT INQUIRE REQUESTMODEL. 262
CEMT INQUIRE RRMS 264
CEMT INQUIRE STATISTICS 265
CEMT INQUIRE STREAMNAME 268
CEMT INQUIRE SYDUMPCODE. 270
CEMT INQUIRE SYSTEM 274
CEMT INQUIRE TASK 280
CEMT INQUIRE TCLASS 287
CEMT INQUIRE TCPIP 288
CEMT INQUIRE TCPIPSERVICE 289
CEMT INQUIRE TDQUEUE 295
CEMT INQUIRE TERMINAL 300
CEMT INQUIRE TRANSACTION. 307
CEMT INQUIRE TRDUMPCODE. 312
CEMT INQUIRE TSMODEL. 315
CEMT INQUIRE TSPOOL 318
CEMT INQUIRE TSQUEUE / TSQNAME. 319
CEMT INQUIRE UOW. 323
CEMT INQUIRE UOWDSNFAIL 328
CEMT INQUIRE UOWENQ 333
CEMT INQUIRE UOWLINK 337
CEMT INQUIRE URIMAP 340
CEMT INQUIRE VTAM 344
CEMT INQUIRE WEB 346
CEMT INQUIRE WEBSERVICE 347
CEMT INQUIRE WORKREQUEST 350
CEMT PERFORM commands 354
CEMT PERFORM CLASSCACHE 354
CEMT PERFORM CORBASERVER 357
CEMT PERFORM DELETSHIPPED. 359
CEMT PERFORM DJAR 359
CEMT PERFORM DUMP|SNAP 360
CEMT PERFORM ENDAFFINITY 361
CEMT PERFORM JVMPOOL 362
CEMT PERFORM PIPELINE 363
CEMT PERFORM RESET 364
CEMT PERFORM SECURITY 364
CEMT PERFORM SHUTDOWN 365
CEMT PERFORM STATISTICS 366

CEMT SET commands 370
CEMT SET AUTOINSTALL 370
CEMT SET AUXTRACE 372
CEMT SET BRFACILITY 373
CEMT SET CLASSCACHE 374
CEMT SET CONNECTION 375
CEMT SET CORBASERVER 381
CEMT SET DB2CONN 383
CEMT SET DB2ENTRY 390
CEMT SET DB2TRAN. 394
CEMT SET DELETSHIPPED 395
CEMT SET DISPATCHER 396
CEMT SET DOCTEMPLATE 398
CEMT SET DSAS 399
CEMT SET DSNAME 399
CEMT SET DUMPDS 404
CEMT SET ENQMODEL 405
CEMT SET FEPI. 406
CEMT SET FILE 406
CEMT SET GTFTRACE 411
CEMT SET HOST 411
CEMT SET INTTRACE 412
CEMT SET IPCONN 413
CEMT SET IRC 417
CEMT SET JOURNALNAME 418
CEMT SET JVMPOOL 419
CEMT SET LIBRARY 420
CEMT SET LINE. 422
CEMT SET MODENAME. 423
CEMT SET MONITOR 424
CEMT SET NETNAME 427
CEMT SET PIPELINE 429
CEMT SET PROCESSTYPE 430
CEMT SET PROGRAM 432
CEMT SET QUEUE 435
CEMT SET STATISTICS 435
CEMT SET SYDUMPCODE 437
CEMT SET SYSTEM 440
CEMT SET TASK 444
CEMT SET TCLASS 445
CEMT SET TCPIP 446
CEMT SET TCPIPSERVICE 447
CEMT SET TDQUEUE 449
CEMT SET TERMINAL 451
CEMT SET TRANSACTION 454
CEMT SET TRDUMPCODE 456
CEMT SET TSQUEUE 457
CEMT SET UOW 458
CEMT SET UOWLINK. 459
CEMT SET URIMAP 460
CEMT SET VTAM 461
CEMT SET WEB. 463
CEMT SET WEBSERVICE 463
CEMT SET WORKREQUEST 464

Chapter 16. CEOT - terminal status 467

Using CEOT in inquiry mode 467
CEOT in set mode 468

Chapter 17. CESF - sign off 471

Chapter 18. CESN - sign on 473
IBM 3270 family display devices 474
Non-3270 family terminal. 475
Command options 475
Signing off with CESN. 476

Chapter 19. CEST—supervisory terminal 477

Chapter 20. CETR - trace control 479
Using component trace options 480
Using Transaction and terminal-related trace options 483
Specifying JVM trace options in CETR. 485
Using CETR HELP panels 486
HELP panels for components 488
HELP panels for terminal and transaction trace 491
HELP panels for JVM trace options 493

Chapter 21. CIDP - inactivate debugging profiles 497

Chapter 22. CIND—in-doubt testing tool 499
Running the CIND indoubt tool 500
Inquiring on the status of CIND 500
Running the in-doubt tool using an EXEC CICS LINK command 501
Setting a single transaction in-doubt 501
CIND Messages 501

Chapter 23. CLDM - CICS load module map 503

Chapter 24. CLER - Language Environment runtime options 505

Chapter 25. CMAC - messages and codes display 507

Chapter 26. CMSG - message switching 511
Sending a message. 511
Canceling a message 512
Message-switching continuation 513
Message delivery 514
Subsystems and terminal systems 514
IBM 3270 display devices 515
IBM 3767 Communication Terminal 515
Sequential terminal 515
CMSG command 515
CMSG command options. 516
Message-switching responses 526
CMSG examples. 533

Chapter 27. CREA - create REQUESTMODELs for enterprise beans . . . 535
Function: 535
Using CREA 536
Starting CREA 536
TranID Specification 537

REQUESTMODEL Specification . . . 541
Results . . . 544
CREC . . . 545

Chapter 28. CRTE - remote transactions . . . 547

Chapter 29. CSFE - terminal and system test . . . 549
Using CSFE terminal test . . . 549
Tracing with CSFE . . . 549
Terminal builder trace . . . 550
Using CSFE debug functions . . . 550
Storage freeze . . . 550
Storage violation trap . . . 551
Global trap/trace program exit and global kernel exit . . . 552

Chapter 30. CSPG - page retrieval . . . 553
Retrieve a page . . . 553
Single keystroke retrieval (SKR) . . . 555
Page retrieval for logical devices . . . 555
Page retrieval and partitions . . . 556
Display information about logical messages . . . 556
Message chaining with CSPG . . . 556
CSPG message termination . . . 559
Copying pages with CSPG . . . 560
Using Autopage with CSPG . . . 560
Displaying routed pages with CSPG . . . 561

Chapter 31. CWTO - write to console operator . . . 563

Chapter 32. DSNC - CICS DB2 transaction DSNC . . . 565

Appendix. List of CICS transactions . . . 567

Bibliography . . . 573
The CICS Transaction Server for z/OS library . . . 573
The entitlement set . . . 573
PDF-only books . . . 573
Other CICS books . . . 575
Books from related libraries . . . 575
Determining if a publication is current . . . 576

Accessibility . . . 577

Index . . . 579

Notices . . . 615

Trademarks . . . 617

Preface

What this book is about

This book tells you how to start the transactions supplied with CICS®. It tells you how to control the CICS system, how to send messages, how to collect statistics, and how to get terminal error information. It is also useful if you are responsible for preparing procedures for people who operate such terminals.

The book also describes how you can use the operating system console as a CICS terminal.

Who should read this book

This book is intended for CICS terminal operators and persons responsible for preparing terminal operating procedures. It is also useful for both system and application programmers.

What you need to know to understand this book

A general knowledge of the concepts and terminology is required.

How to use this book

This book is mainly for reference. After a short introduction, the rest of the book is arranged in alphabetic order of the transaction identifier. An index is provided to pinpoint individual transactions and keywords.

Some CICS transactions are described in greater detail in other CICS books. These include:

Transaction	CICS book
CBAM	*CICS Business Transaction Services*
CDBC	*CICS IMS Database Control Guide*
CDBI	*CICS Data Areas*
CEBR	*CICS Application Programming Guide*
CECI	*CICS Application Programming Guide*
CECS	*CICS Application Programming Guide*
CEDA	*CICS Resource Definition Guide*
CETR	*CICS Problem Determination Guide*
CRTE	*CICS RACF Security Guide*
CSFE	*CICS Problem Determination Guide*
DSNC	*CICS DB2 Guide*

In these cases, this book contains a brief description of the transaction and indicates where further information is to be found.

Front end programming interface

Information about the Front End Programming Interface (FEPI) feature is not given in this book. See the *CICS Front End Programming Interface User's Guide* for relevant information.

© Copyright IBM Corp. 1982, 2011

Notes on terminology

- CICS refers to IBM® CICS Transaction Server for z/OS®, Version 3 Release 2
- VTAM® refers to IBM ACF/VTAM
- IMS™ refers to IBM IMS/ESA®

Summary of changes

This book is based on the CICS Supplied Transactions for CICS Transaction Server for z/OS, Version 2 Release 1. Changes from that edition are marked by vertical bars in the left margin. Minor amendments that have been made since the most recent published version are marked by # symbols in the left margin.

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

For most items in each of the lists above, there is a link to the part of the book where there is more detail.

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

For information about changes that have been made in CICS Transaction Server for z/OS, Version 3 Release 2, please refer to *What's New* in the information center, or the following publications:

- *CICS Transaction Server for z/OS Release Guide*
- *CICS Transaction Server for z/OS Migration from CICS TS Version 3.1*
- *CICS Transaction Server for z/OS Migration from CICS TS Version 2.3*
- *CICS Transaction Server for z/OS Migration from CICS TS Version 2.2*
- *CICS Transaction Server for z/OS Migration from CICS TS Version 1.3*

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

The more significant changes for this edition are:

- Technical changes :
  - –

    Because of the removal of support for HPJ (Hot-pooling) for Java program objects, the following items are removed:
    - The HOTPOOLING attribute is removed from the CEMT INQUIRE PROGRAM command.
    - The HOTPOOL and NOTHOTPOOL attributes are removed from the CEMT SET PROGRAM command.
    - The ACTHPTCBS and MAXHPTCBS attributes are removed from the CEMT INQUIRE DISPATCHER command.
    - The MAXHPTCBS attribute is removed from the CEMT SET DISPATCHER command.
- New commands :
  - There is a new transaction called CCRL that creates and updates certificate revocation lists. See Chapter 4, “CCRL - certificate revocation lists transaction,” on page 59 for details.
- New options:
  - –
- Structural changes :
  - –

© Copyright IBM Corp. 1982, 2011

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

The more significant changes for this edition are:

- Technical changes :
  - Chapter 2, “The application debugging profile manager,” on page 17 has been added.
  - Chapter 21, “CIDP - inactivate debugging profiles,” on page 497 has been added.
- New commands :
  - “CEMT INQUIRE WORKREQUEST” on page 350 added to CEMT
  - “CEMT SET WORKREQUEST” on page 464 added to CEMT
  - “CEMT INQUIRE CLASSCACHE” on page 151 added to CEMT
  - “CEMT SET CLASSCACHE” on page 374 added to CEMT
  - “CEMT PERFORM CLASSCACHE” on page 354 added to CEMT
  - “CEMT INQUIRE JVM” on page 229 added to CEMT
- New options:
  - DEBUGTOOL added to CEMT INQUIRE SYSTEM
  - DEBUG added to CEMT SET SYSTEM
  - NODEBUG added to CEMT SET SYSTEM
- Structural changes :
  -

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

The more significant changes for this edition are:

- Technical changes :
  - In “CEMT INQUIRE DB2CONN” on page 167, the definitions of parameters DB2ID, PRIORITY, TCBLIMIT and TCBS have changed.
  - In “CEMT INQUIRE DB2ENTRY” on page 174, the definition of the PRIORITY parameter has changed.
  - In “CEMT SET DB2CONN” on page 383, the definitions of parameters DB2ID, PRIORITY, TCBLIMIT and TCBS have changed.
  - In “CEMT SET DB2ENTRY” on page 390, the definition of the PRIORITY parameter has changed.
  - In “CEMT INQUIRE TCPIPSERVICE” on page 289, there are changes to AUTHENTICATE and the description of its CVDA values.
  - There are new transactions CREA and CREC, see Chapter 27, “CREA - create REQUESTMODELs for enterprise beans,” on page 535 for details.
- New options:
  - DB2GROUPID added to CEMT INQUIRE DB2CONN
  - RESYNCMEMBER added to CEMT INQUIRE DB2CONN
  - PLAN added to CEMT INQUIRE DB2TRAN
  - PLANEXITNAME added to CEMT INQUIRE DB2TRAN
  - SCAN added to CEMT PERFORM CORBASERVER
  - DB2GROUPID added to CEMT SET DB2CONN
  - RESYNCMEMBER added to CEMT SET DB2CONN

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

The more significant changes for this edition are:

- HELP panel removed from CEBT
- New commands:
  - "CEMT INQUIRE BEAN" on page 144
  - "CEMT INQUIRE CORBASERVER" on page 163
  - "CEMT INQUIRE DJAR" on page 183
  - "CEMT INQUIRE JVMPOOL" on page 231
  - "CEMT PERFORM CORBASERVER" on page 357
  - "CEMT INQUIRE UOWENQ" on page 333
  - "CEMT DISCARD commands" on page 132
  - "CEMT DISCARD commands" on page 132
  - "CEMT SET CORBASERVER" on page 381
  - "CEMT SET JVMPOOL" on page 419
- New chapters:
  - Chapter 24, "CLER - Language Environment runtime options," on page 505
  - Chapter 28, "CRTE - remote transactions," on page 547
- New options
  - HOTPOOLING was added to CEMT INQUIRE PROGRAM. (This option was removed for CICS Transaction Server for z/OS, Version 3 Release 1.)
  - JVMPROFILE added to CEMT INQUIRE PROGRAM
  - BEANNAME added to CEMT INQUIRE REQUESTMODEL
  - CORBASERVER added to CEMT INQUIRE REQUESTMODEL
  - INTERFACE added to CEMT INQUIRE REQUESTMODEL
  - INTFACETYPE added to CEMT INQUIRE REQUESTMODEL
  - MODULE added to CEMT INQUIRE REQUESTMODEL
  - OPERATION added to CEMT INQUIRE REQUESTMODEL
  - TYPE added to CEMT INQUIRE REQUESTMODEL
  - ACTSOCKETS added to CEMT INQUIRE TCPIP
  - MAXSOCKETS added to CEMT INQUIRE TCPIP
  - AUTHENTICATE added to CEMT INQUIRE TCPIPSERVICE
  - DNSGROUP added to CEMT INQUIRE TCPIPSERVICE
  - DNSSTATUS added to CEMT INQUIRE TCPIPSERVICE
  - GRPCRITICAL added to CEMT INQUIRE TCPIPSERVICE
  - OTSTIMEOUT added to CEMT INQUIRE TRANSACTION
  - OTSTID added to CEMT INQUIRE UOW
  - HOST added to CEMT INQUIRE UOWLINK
  - CORBASERVER added to CEMT PERFORM STATISTICS
  - JVMPOOL added to CEMT PERFORM STATISTICS
  - REQUESTMODEL added to CEMT PERFORM STATISTICS
  - TCPIP added to CEMT PERFORM STATISTICS
  - HOTPOOL was added to CEMT SET PROGRAM. (This option was removed for CICS Transaction Server for z/OS, Version 3 Release 1.)
  - NOTHOTPOOL was added to CEMT SET PROGRAM. (This option was removed for CICS Transaction Server for z/OS, Version 3 Release 1.)
  - MAXSOCKETS added to CEMT SET TCPIP
  - DNSSTATUS added to CEMT SET TCPIPSERVICE
  - NOUCTRAN added to CEOT IN SET MODE
  - TRANIDONLY added to CEOT IN SET MODE
  - UCTRAN added to CEOT IN SET MODE
- New values
  - OTSCOORD added to WAITCAUSE in CEMT INQUIRE UOW

Chapter 1. Using CICS supplied transactions

CICS provides operations, usually initiated from terminals, called *transactions*, each of which involves the use of CICS tables, programs, and internal services. The following sections describe the transactions that are supplied by CICS and that have an operator interface.

CICS transactions have identification codes that start with "C" and are 4 characters long; for example, CEMT. The following table shows the transactions that are described in detail. For a complete list of CICS transactions, including those that do not have an operator interface, see the "List of CICS transactions," on page 567.

Function	Name	Page
CICS business transaction services (BTS) browser	CBAM	Chapter 3, "CBAM - BTS browser," on page 57
Command-level interpreter	CECI, CECS	Chapter 10, "CECI - command-level interpreter," on page 91
Create and update certificate revocation lists (CRLs)	CCRL	Chapter 4, "CCRL - certificate revocation lists transaction," on page 59
Create REQUESTMODELs for enterprise beans	CREA, CREC	Chapter 27, "CREA - create REQUESTMODELs for enterprise beans," on page 535
Database control inquiry	CDBI	Chapter 6, "CDBI - database control inquiry," on page 65
Database control interface	CDBM	Chapter 7, "CDBM - database control interface," on page 67
Database control menu	CDBC	Chapter 5, "CDBC - database control menu," on page 61
Emergency Use	CEKL	Chapter 13, "CEKL—master terminal (emergency use)," on page 115
Execution diagnostic facility	CEDF, CEDX	Chapter 12, "CEDF and CEDX - the execution diagnostic facility," on page 109
In-doubt testing tool	CIND	Chapter 22, "CIND—in-doubt testing tool," on page 499
Master terminal	CEMT	Chapter 15, "CEMT—master terminal," on page 123
Master terminal (alternate CICS)	CEBT	Chapter 9, "CEBT - master terminal (alternate CICS)," on page 79

© Copyright IBM Corp. 1982, 2011

Function	Name	Page
Messages and codes display	CMAC	Chapter 25, “CMAC - messages and codes display,” on page 507
Message switching	CMSG	Chapter 26, “CMSG - message switching,” on page 511
Page retrieval	CSPG	Chapter 30, “CSPG - page retrieval,” on page 553
Remote transactions	CRTE	Chapter 28, “CRTE - remote transactions,” on page 547
Resource definition online (RDO)	CEDA, CEDB, CEDC	Chapter 11, “CEDA - resource definition online,” on page 95
Sign off	CESF	Chapter 17, “CESF - sign off,” on page 471
Sign on	CESN	Chapter 18, “CESN - sign on,” on page 473
Supervisory terminal	CEST	Chapter 19, “CEST—supervisory terminal,” on page 477
Temporary-storage browse	CEBR	Chapter 8, “CEBR - temporary storage browse,” on page 75
Terminal status	CEOT	Chapter 16, “CEOT - terminal status,” on page 467
Trace control	CETR	Chapter 20, “CETR - trace control,” on page 479
Terminal and system test	CSFE	Chapter 29, “CSFE - terminal and system test,” on page 549
Write to console operator	CWTO	Chapter 31, “CWTO - write to console operator,” on page 563

In general, you start a CICS transaction by entering its transaction identifier (for example, CEMT). The transaction identifier is used by CICS to identify the programs that handle the specified transactions, and to establish a task to process them.

If you use an IBM 3270 system display or similar display device that has the appropriate features installed, you can also start a transaction by a program function (PF) key or program attention (PA) key, by an operator identification card reader, by a magnetic slot reader, or by a light pen. For other types of terminals or subsystems, see the appropriate CICS/OS/VS subsystem guide.

You may wish to apply a CICS-supplied upgrade, but are using modified versions of one or more CICS-supplied transactions or of the CICS-supplied calling programs that handle CICS-supplied transactions. After you have first copied them to differently named groups, you must replace these private versions from the

upgraded CICS-supplied version and modify them afresh to ensure that the necessary upgrade changes are carried out. Failure to do this can lead to unpredictable results.

This chapter describes:

- "How to start or stop a transaction"
- "Syntax notation and conventions used"
- "Terminal operation" on page 5
- "Using the system console" on page 10
- "Using TSO consoles" on page 13

How to start or stop a transaction

You start a CICS transaction by pressing the CLEAR key to clear the screen, and entering the transaction identifier, either by itself or followed by data, on the command line of the screen. The command line is a single line, usually at the top of the screen.

You can type the transaction identifier by itself and follow the prompts until a complete transaction command is built up, or you can type the complete transaction command on the command line. If you do not enter enough information, or if the information you enter is wrong, you are prompted to complete or correct your input.

For example, in the following transaction, CEMT is the transaction identifier and the additional data is INQUIRE PROGRAM(PROG1).

```
CEMT INQUIRE PROGRAM(PROG1)
```

When the transaction starts, it processes the additional data. At the completion of this transaction, you get the following message:

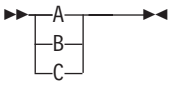
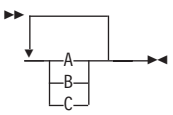
```
STATUS: SESSION ENDED
```

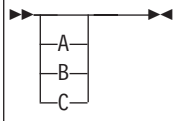
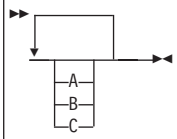
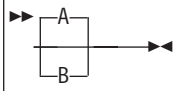
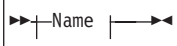
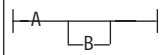
After a transaction has completed, press the CLEAR key to clear the screen in readiness for the next transaction. You can cancel any request by typing CANCEL on the command line.

Syntax notation and conventions used

Each command has a syntax box to show you what options there are.

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

Symbol	Action
	A set of alternatives—one of which you ***must*** code.
	A set of alternatives—one of which you ***must*** code. You ***may*** code more than one of them, in any sequence.

Symbol	Action
	A set of alternatives—one of which you ***may*** code.
	A set of alternatives — any number (including none) of which you may code once, in any sequence.
	Alternatives where **A** is the default.
 **Name:** 	Use with the named section in place of its name.
Punctuation and uppercase characters	Code exactly as shown.
Lowercase characters	Code your own text, as appropriate (for example, `name`).

Minimum abbreviation of keywords

In general, the CICS transactions accept as few characters of a keyword as needed to identify it within the request.

For example, in a CEMT INQUIRE TASK command, you could use TASK, TAS, or TA to uniquely identify TASK. T alone, however, could be confused with TCLASS, TERMINAL, TRACE, or TRANSACTION.

In the syntax displays on your screen (unless your terminal is uppercase only), and in most cases that describe the supplied transactions, the minimum permitted abbreviation is given in uppercase characters, the remainder in lowercase.

Uppercase input to transactions

In general, most CICS-supplied transactions accept only uppercase input. If UCTRAN=YES has been specified in the terminal definition, all lowercase characters, even those enclosed within single quotation marks, are translated to uppercase.

If you have to specify UCTRAN=NO for your terminal, you have to ensure that the group specified for your terminal refers to a profile that will carry out uppercase translation.

CICS provides a PROFILE definition, DFHCICSP, in the DFHSTAND group in the CICS system definition (CSD) file. This profile is identical to DFHCICST except that it specifies UCTRAN(YES) instead of UCTRAN(NO).

The new profile is used by the CICS-supplied page retrieval transaction, CSPG. The new profile, together with changes in the task-attach routine and the page retrieval program, enables CICS to perform uppercase translation at the transaction level for BMS paging.

This allows users of terminals that are defined with uppercase translation switched off to use the page retrieval function without having to enter paging commands in upper case. Assigning a new profile for CSPG means that all data entered on the retrieval command (defined by the PGRET system initialization parameter) and the purge command (defined by the PGPURGE system initialization parameter) is translated to uppercase.

If a user's terminal is defined with UCTRAN(YES), the new profile has no effect because all terminal input is translated to uppercase anyway.

Terminal operation

A CICS system makes provision for the following classes of operators:

- A *terminal operator* who can use a terminal to perform routine transactions that cause application programs to be processed. You can use a small selection of CICS transactions. For example, you can inquire about, or change, the status of your own terminal.
- A *supervisory terminal operator* who can perform all the duties of a terminal operator, in addition to supervising other operators within a functional group. Your operator security code gives you access to the supervisory terminal transaction, CEST, with which you can monitor and control some of the system resources used by your group.
- A *master terminal operator* who can monitor and control resources in a CICS system. Your operator security code gives you access to the master terminal transaction, CEMT, with which you monitor and control the system resources. Internal security checking may limit the range of resources under your control.

$ (the dollar symbol)

In the character sets given in this information, the dollar symbol ($) is used as a national currency symbol and is assumed to be assigned the EBCDIC code point X'5B'. In some countries a different currency symbol, for example the pound symbol (£), or the yen symbol (¥), is assigned the same EBCDIC code point. In these countries, the appropriate currency symbol should be used instead of the dollar symbol.

Operator security

The transactions you can initiate are defined by your profile in the external security manager (ESM) database, which is normally provided when you sign on using the CESN transaction.

Generally, the master terminal operator has access to all CICS-supplied transactions, the supervisory terminal operator has access to a subset, and the terminal operator has access to very few transactions.

The system programmer is responsible for allocating operator security codes to restrict the use of particular transactions. For more information, see the relevant system administration guide for the ESM you are using. For example, if you are using RACF, see the *z/OS Security Server RACF Security Administrator's Guide*.

Terminal operator

To operate the system, you normally first sign on to the system and, as a minimum, enter your user ID and your password.

During signon, the information you enter is used by CICS to establish priorities and your ESM profile for the transactions that you might want to use later. When you have signed on, you have access to those transactions defined in your ESM profile.

After you have signed on, you can enter only specific transaction identifiers. Be aware of error messages that might be generated by the transactions you start, and the corrective action that you must take. In addition to error messages, be aware of other messages that CICS might transmit to your terminal.

You need to know the terminal identifiers of other terminals with which you want to communicate.

Supervisory terminal operator

A supervisory terminal operator is the supervisor of any part of the system for which group control is desired. You are responsible for supervising, and keeping operational, groups of terminals defined in one or more terminal list tables (TLTs). You do this using the supervisory terminal transaction, CEST.

Your responsibilities can be thought of as a subset of those of the master terminal operator, and you should be aware of those functions that are not available to the terminals under your supervision. You also should be aware of, and understand, the procedure for changing the status of each terminal.

You should know the identifiers of all terminals and operators under your supervision. The terminal identifiers are defined in one or more CICS terminal list tables (TLTs). Individual TLTs can be identified by a 1-or 2-character suffix you enter as the SUPRID and CLASS(value) keywords of the CEST transaction.

When you use the CEST command for *all* terminals defined in a TLT, you have to specify the SUPRID keyword followed by the 2-character identifier of that TLT.

A subset of those terminals can be grouped together as a *class*, and can be defined as such in a different TLT. For information about defining the TLT itself, see the *CICS Resource Definition Guide*. You can specify that class by means of the CLASS(value) keyword of the CEST SET TERMINAL command, where 'value' is the suffix that identifies the TLT in which the class of terminals has been defined.

Alternatively, you can name one or more terminals in the CEST SET TERMINAL(value) command itself.

If you frequently want to restrict a CEST command to a subset of your terminals, and have defined another TLT identifying that subset, you then have to use the CEST SET TERMINAL SUPRID(value) keyword to refer to the "main" TLT, followed by the CLASS(value) keyword to refer to the specific TLT containing the subset.

So, for example, if you have defined terminals S202, S203, S204, and S205 in DFHTLTAB and you want to issue a CEST command that sets *all* those terminals out of service, you issue the following command:

```
CEST SET TERMINAL SUPRID(AB) OUTSERVICE
```

If, on the other hand, you want to restrict your command(s) to terminals S202 and S204, for example, you could define these two terminals in another TLT—DFHTLTCD, say—and issue the following command:

```
CEST SET TERMINAL SUPRID(AB) CLASS(CD) OUTSERVICE
```

Alternatively, of course, you could issue the following command:

```
CEST SET TERMINAL(S202,S204) SUPRID(AB) OUTSERVICE
```

Unless otherwise stated, the information in this book about the supervisory terminal and the CEST transaction applies only to a single CICS system, regardless of whether it is connected to another CICS system through ISC or MRO.

Master terminal operator

The master terminal operator controls system components using the master terminal transaction, CEMT. With this transaction, the master terminal operator can dynamically change the system control parameters.

Although the transaction can be started at any valid IBM 3270 family display device or equivalent terminal, or from the operating system console, its use is intended to be limited to a person known as the *master terminal operator*. Starting a transaction from the operating system console is described in “Using the system console” on page 10.

The control permitted through CEMT allows you, the master terminal operator, to improve performance by changing the system control parameters in the day-to-day operation of the system. In addition to system control, you have prime responsibility for administering the terminal facilities of the system.

By using the routing transaction (CRTE), you can also be a master terminal operator for multiple connected CICS systems.

As the master terminal operator, you can access all terminal and supervisory terminal transactions. In addition, however, you must be familiar with all the procedures associated exclusively with the master terminal. You must be aware of which terminals and operators can access CICS at any given time, and of the identifiers by which they are known to CICS.

For example, when inquiring about terminals, you can specify a class of terminals or a list of terminals. A class of terminals is specified by the CLASS(value) keyword, where ‘value’ is the 1-or 2-character suffix of the related terminal list table (DFHTLTxx). A list of terminals is specified by a series of terminal identifiers following the CEMT SET TERMINAL(t1,t2,...,) command, where t1, t2, are terminal identifiers. See “List of resource identifiers” on page 128 for more information.

For MRO and LUTYPE6.1 connections, you must know the identifier of each parallel session, and specify this identifier when operating on the session.

For LUTYPE6.2 (APPC) connections, you must know the modename of each set of parallel sessions, and specify this modename when operating on the modegroup.

Your use of the master terminal transaction is restricted by entries in the signon table and in the installed transaction definitions. These entries are the responsibility of the system programmer.

During long periods of continuous operation, you can, at intervals, read out and reset the statistics counters. The volume of activity in your system determines how often you should do this.

When the system has satisfactorily completed its response to a command, the time and date are printed or displayed at your terminal, as follows:

```
TIME=hh.mm.ss DATE=mm.dd.yy
```

where time is in hours, minutes, and seconds, and date is in months, days, and years, or in the form specified by the DATFORM system initialization parameter. For brevity this final message has been deleted from all further examples.

Unless otherwise stated, the information about the master terminal and its transactions applies only to a single CICS system, regardless of whether it is connected to another CICS system through ISC or MRO.

MRO and ISC support

Multiregion operation (MRO) and intersystem communication (ISC) allow the sharing of resources between more than one CICS region. Thus a user at a terminal assigned to one CICS region can run transactions in connected regions, and can access resources - files, for example - in other regions.

It is also possible for a transaction running in one region to communicate with a transaction running in another region, thus sharing the processing workload.

Except for experiencing longer response times, you should not be aware that MRO or ISC processes are being used.

BMS partitions

When you use display devices that support BMS partitions, make sure that you understand:

- How to use the SCROLL, PARTITION JUMP, CLEAR, and CLEAR PARTITION keys
- The concept of the active partition
- The meanings of the partition-related indicator symbols that can appear on a display screen.

For information about BMS partitions, see the *CICS Application Programming Guide*.

CLEAR key

The CLEAR key clears all partitions from the display, and sets the terminal to ‘base’ state. The next BMS output request recreates the partitions (but does not restore their contents), using the application partition set.

The CLEAR and CLEAR PARTITION keys cannot be used interchangeably when an existing CICS transaction is run in a single explicit partition.

Partitions and the execution diagnostic facility

The execution diagnostic facility (EDF), invoked by CEDF, is not available in single-screen mode on a terminal in partitioned state. EDF must be used in dual-screen mode for debugging application programs that use partitions.

Partitions and the command interpreter

The CICS command interpreter, invoked by CECI or CECS, cannot be used to process commands that refer to partitions. This is because the command interpreter display cannot be restored after the screen has been partitioned.

PA1 print key

The PA1 print key is not supported from a terminal in partitioned state.

Routing and multiple partitions

Routed messages can be directed to a terminal, including the transaction terminal, which supports partitions. However, such messages reset the terminal to 'base' state.

Terminal paging

When a BMS logical message is saved in CICS temporary storage, CICS also saves the application partition set. This partition set is loaded onto the target terminal before any pages are delivered. CICS builds a separate page for each partition, and overflow occurs on a partition basis.

Page retrieval

Terminal-operator page-retrieval commands operate on a partition basis.

When a page-retrieval or page-copy command is entered in a partition, it implicitly refers to pages in that partition. If single-keystroke retrieval is used, the retrieval command applies to the partition containing the cursor when the PF key is pressed. The first page for a partition is displayed initially in the viewport.

Message chaining

CICS retains a current partition for each level of page chaining. This is initially the default partition for that partition set. Page-retrieval commands entered on a cleared screen, or page-retrieval commands for a chaining level other than the one being displayed, refer to the current partition for the target chaining level. The current partition is reset to the partition in which the last terminal-operator command was entered.

CICS retains the current page for each partition in the partition set. This is initially the first page. The current page is redisplayed in each partition in the following circumstances:

- For the initial display when the BMS paging program is first invoked
- Following erasure of the terminal partition set caused by pressing the CLEAR key
- Following page retrieval for a different page-chaining level
- Following page purge for a different page-chaining level.

Copying pages

BMS page copy operates on a partition basis (not a screen or partition set basis). BMS page copy copies a page from a partition to any terminal in 'base' state. You cannot copy a page from a partition to another partition on the same or another terminal.

Message termination

When you terminate a message, the entire logical message (that is, all pages in all partitions) is purged, irrespective of the partition in which you entered the purge command.

The response to a page query request is displayed on a cleared, unpartitioned screen.

Error messages

Most error messages relating to invalid paging commands are displayed with an erase or write in the partition in which you entered the command. Other error messages unrelated to any particular partition (such as those relating to invalid message identifiers) are displayed on a cleared unpartitioned screen.

Using the system console

Console support makes it possible for a terminal to be both an operating system console and a CICS master terminal.

If multiple console support (MCS) is in use, you can define each console to CICS as a separate terminal, and all consoles can communicate with CICS simultaneously.

You can use any operating system console as a CICS terminal if it has been specified as such on the CONSOLE keyword of the CEDA DEFINE TYPETERM command. If this has not been done, you get the following message when you try to use the console:

```
DFHAC2015 This console has not been defined to CICS.
```

and your input is ignored.

All consoles that have been defined as CICS terminals can use automatic transaction initiation (ATI), and can receive messages from other terminals and consoles, as well as from CICS transactions.

In a system that has consoles and VTAM terminals, a console can remain active when CICS and VTAM are disconnected from each other. You can use the console to make or break the CICS-VTAM connection without CICS being terminated.

Use the MODIFY and REPLY commands to start the CICS-supplied transactions from an operating system console.

In addition to the MODIFY and REPLY commands, the system programmer should consider use of the CONTROL, DISPLAY, START, and VARY commands when preparing console operator procedures. For information on these commands and other system details, see the *OS/390 MVS System Commands* manual.

Rules for console entry

Commands typed at a console are translated to uppercase, except for characters enclosed within single quotation marks (' '), which remain unchanged. The occurrence of a literal single quotation mark must be indicated by a pair of single quotation marks (''), for example:

```
'Please phone Mr O''Neill'.
```

If UCTRAN=YES has been specified in the terminal definition, all lowercase characters, even those enclosed within single quotation marks, are translated to uppercase.

MODIFY command

You start a CICS transaction from a console by using the MODIFY command, as follows:

```
MODIFY ident,datastring
```

You can abbreviate the MODIFY command to F.

`ident` can be any of the following:

- The name of the job used to start CICS, when it is started by a job stream.
- The name of the procedure used to start CICS, when it is started by an MVS START command, for example:

  ```
  START procedure_name
  ```

  where “procedure_name” is the ident value.
- The task identifier that was used to qualify the procedure name, for example:

  ```
  START procedure_name.taskid
  ```

  where “taskid” is the ident value. This is likely to be used where the same procedure is started more than once.

`datastring` is a string of data, starting with a CICS transaction identifier.

For example, to start transaction CEBT on the CICSA system from the console, type:

```
MODIFY CICSA,CEBT PERFORM TAKEOVER
```

You can type more than one MODIFY command at a console; each is processed in order of entry.

A CICS transaction can issue terminal control READ, WRITE, or CONVERSE commands to communicate with a console operator. WRITE and CONVERSE transmit application program messages, but READ simply produces a prompt, incorporating message ‘DFH4200A’, as follows:

```
@nn DFH4200A jjjjjjjj tttt
```

where:

nn is the number (generated by the operating system) that you must use in your reply to the prompt. Messages from a transaction that uses CONVERSE commands also contain this number.

jjjjjjjj is the jobname of CICS in the operating system.

tttt is the transaction identifier of the CICS transaction that has issued the READ command.

REPLY command

You (the console operator) must respond to each prompt by using the REPLY command, which you can type at either the prompted console or the master console:

```
R[EPLY] nn,datastring
```

where `nn` is the number of the prompt to which you are replying, and `datastring` is your reply.

If a transaction is purged while it is awaiting a reply from the operator, the reply is canceled.

You should note that messages to the console can become interspersed with messages from the operating system and from other regions, making them difficult to read. In extreme cases, parts of lengthy messages can 'scroll off' the console screen before they have been read.

Example of a conversation using CONVERSE

```
modify job002,serv  1

@17 FAULT TYPE?   2

r 17,elec   3

MESSAGE HAS BEEN SENT
```

1 MODIFY command specifying that transaction "serv" is to be started; this transaction sends messages to service groups supporting the installation.

2 The transaction response produced by a CONVERSE command and relayed by the operating system.

3 Your reply that the fault is an electrical one.

Example of a conversation using WRITE/READ

```
modify job002,usid   1

USER SIGNON ID=?   2

@25 DFH4200A JOB002 USID  3

r 25,accts1   4

USER'S NAME: J. SMITH   5
USER'S TEL. NO.: 88999   6
```

1 MODIFY command specifying that transaction "usid" is to be started. This transaction provides information about the user identified by "usid".

2 Application-program message produced by a WRITE command.

3 System message produced by a READ command.

4 Your reply.

5 Transaction message, produced by a WRITE command, giving the requested information. No reply is needed.

6 Another transaction message, produced by a WRITE command, giving more requested information. Again, no reply is needed.

Using TSO consoles

A TSO session can be used to input CICS commands. This has several advantages:

- It removes the MVS limitation of 99 consoles.
- It supports additional device types, such as the IBM 3290.
- It supports remote operation.

The console ID is 4 bytes, only one of which is used for locally connected consoles. TSO and JES3 consoles use all 4 bytes. In addition, each console in a sysplex has an 8-byte name.

Using the KILL option for purging transactions

It is possible for a looping CICS task or shortage of CICS resources to bring all normal processing in a CICS region to a halt.

In such a situation, it could become impossible for you to cancel the task or tasks using any of the standard CICS mechanisms, for example, PURGE or FORCEPURGE. In other cases, it might not even be possible to list the active tasks in order to investigate the problem. The only way for you to recover from such a situation is to recycle the CICS region. The purpose of this new function is to enhance your ability to remove such tasks more quickly from a CICS region with the minimum impact on the integrity of your systems.

If you can identify the task that is causing the problem you should try to use the PURGE or FORCEPURGE options of the CEMT SET TASK command. Both PURGE and FORCEPURGE have been significantly enhanced and using these options may well enable you to remove the task retaining system and data integrity.

If neither PURGE or FORCEPURGE respond within a reasonable time, you can use the new KILL option that has been added to the CEMT SET TASK, CEMT SET CONNECTION, and CEMT SET TERMINAL commands. The KILL option allows you to cancel tasks that could not have been cancelled using the PURGE or FORCEPURGE options. The KILL option is also supported on the equivalent EXEC CICS commands.

It may be impossible for you to use normal methods, such as CEMT or CICSPlex SM to list the active tasks in the system and hence to determine which task or tasks should be cancelled. To overcome this a new transaction, CEKL, has been added.

The CEKL transaction differs from other CICS supplied transactions in that you do not have to install definitions for it. Also you can invoke it only from an operating system console, which does not have to be defined to CICS. The CEKL INQUIRE TASK command allows you to list information about tasks in the CICS region; for example, all tasks that are suspended. The CEKL SET TASK command allows you to cancel a selected task using the PURGE, FORCEPURGE or KILL option. The

CEKL transaction forms part of the CICS console interface and runs on a dedicated CICS internal TCB which enables commands to be processed even when the rest of CICS is unresponsive.

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.

Using CEMT to terminate a task

Having identified the problem task, you can then use the normal process of attempting to remove the task from your system using either PURGE or FORCEPURGE through the CEMT SET TASK (value) PURGE transaction.

Initially you can track the state of the task to see if it has successfully removed. Two things may have happened:

- The task was not purged within an acceptable length of time or
- The CEMT transaction did not respond.

If the task could not be purged but CEMT is available, try using the new KILL option. You must try FORCEPURGE before using KILL. See “Consequences of using KILL” on page 15 before using this option.

If CEMT is unavailable, consider using the new CEKL command, see “Using the console to terminate a task.”

Using the console to terminate a task

You can identify problems in your system using the CEMT transaction or CICSPlex SM. However, if these are not available or not running (because the QR TCB is looping or suspended), you can use the new command, CEKL:

- To identify the problem tasks
- To remove the tasks by executing the PURGE, FORCEPURGE, or KILL option
- To monitor the effects of attempting to remove the tasks from your system.

For full details about CEKL, see Chapter 13, “CEKL—master terminal (emergency use),” on page 115.

Identifying tasks using CEKL

You can use CEKL INQUIRE TASK in a number of ways:

- CEKL INQUIRE TASK returns a list of tasks on your system, see Chapter 13, “CEKL—master terminal (emergency use),” on page 115 for more information.
- You can restrict the list of tasks by specifying the TRANCLASS or TRANSID option.
- You can also restrict the list by specifying: DISPATCHABLE, or RUNNING, or SUSPENDED, for example, using DISPATCHABLE returns all the dispatchable tasks in your system.

Removing tasks using CEKL

CEKL SET TASK(value) has three options: PURGE, FORCEPURGE or KILL. The preferred sequence of issuing the commands is as follows:

1. Try PURGE first if the task is purgeable.
2. If the task is not purged in a reasonable time, try using the FORCEPURGE option.

3. If the task is not purged in a reasonable time, try using the KILL option.

Note: Using CEKL you can request a task to be killed without having first tried PURGE or FORCEPURGE, see "Consequences of using KILL."

Monitoring tasks using CEKL

CEKL INQUIRE TASK(*value*) returns the details of a single task. This is useful to monitor a task after you have requested it to be purged, forcepurged or killed.

Consequences of using KILL

WARNING: This action may result in unpredictable effects, including overwriting of data within the CICS region or abnormal termination of the CICS region. It should only be used as a last resort.

Following the use of the KILL option, an abend code is issued that reports the protective state of the task (purge or forcepurge protection) when it was killed. Regardless of the state of the transaction KILL cannot guarantee any data or system integrity.

Using CEMT (or equivalent SPI commands), you are not allowed to kill a task without having first attempted to forcepurge it.

You can use CEKL to kill the task without having first forcepurged it. Use CEKL is very much the option of last resort.

Removing tasks may have an effect on the integrity of your system, Table 1 shows the possible consequences of using purge, forcepurge or kill.

Table 1. Data and system integrity

	System integrity	Data integrity
PURGE	YES	YES
FORCEPURGE	YES	NO
KILL	NO	NO

CICS ignores a request to purge a task if the task is marked non-purgeable.

CICS may ignore or delay processing a request because of a risk of damaging CICS system integrity, for example, because I/O may still be in progress to a buffer owned by the task, or because recovery processing would be unable to complete.

Chapter 2. The application debugging profile manager

Use the application debugging profile manager to manage your debugging profiles.

You can use it to perform the following functions:

- Display a list of debugging profiles
- View the contents of profiles
- Create new profiles
- Modify existing profiles
- Copying existing profiles
- Delete debugging profiles
- Activate and inactivate profiles
- Associate a debugging display device with active profiles

The application debugging profile manager has two user interfaces:

- A Web browser interface. For more information, see “Using the application debugging profile manager Web interface.”
- A terminal interface (the CADP transaction). For more information, see “Using the application debugging profile manager 3270 interface” on page 37.

A utility transaction (CIDP) lets you inactivate all debugging profiles in the system. For details see Chapter 21, “CIDP - inactivate debugging profiles,” on page 497.

Using the application debugging profile manager Web interface

Start the application debugging profile manager by typing its URL in your Web browser. The URL that you enter will depend upon how CICS Web support is configured. Your system administrator will tell you the URL to use for your system. CICS displays the “List profiles” page.

For example, if your Web browser connects directly to CICS, and your system is configured to use the sample analyzer program DFHWBADX, the URL is

`http://mvs_address:port/CICS/CWBA/dfhdpwb`

If URIMAP definitions are being used to manage requests, the URL is as defined in the URIMAP definition that references program DFHDPWB.

Configuring access to the application debugging profile manager Web interface

To provide access to the Web interface, you can use either an analyzer program, or URIMAP definitions, as part of your CICS Web support architecture.

The *CICS Internet Guide* explains the architecture elements that are used to give Web clients access to CICS applications.

Only one URL is required. The user accesses different pages by selecting interface elements on the pages that the program provides, beginning with the “List profiles” page.

- If you want to use the sample analyzer program DFHWBADX, or an analyzer program with similar function, to handle requests for the application debugging

© Copyright IBM Corp. 1982, 2011

profile manager, configure the path component of the URL to give CICS the information it needs to run the program. The path `/CICS/CWBA/dfhdpwb` tells the analyzer that:

  – No converter program is required (`CICS` indicates this).
  – The default transaction ID CWBA is used.
  – The program DFHDPWB, which is the application debugging profile manager, is to be run.

- If you want to use URIMAP definitions to handle requests for the application debugging profile manager, you need to set up two URIMAP definitions.
  – One URIMAP definition matches the URL that is entered by the user to start the application debugging profile manager's Web interface, and maps to the program DFHDPWB.
  – The second URIMAP definition is used to run the CICS program DFHADWB1, which displays graphics.

  The second URIMAP definition is required because when a graphic is wanted, the HTML pages for the Web interface use the tag `img src=`, so that the Web browser makes a request for the graphic and then displays what it receives. The `img src=` tag specifies the name of the graphics program, and the Web browser makes the request using the path component of the URL that was originally used to start the Web interface, followed by the name of the graphics program. A URIMAP definition is needed to map this new request to the graphics program.

The following sample URIMAP definitions could be used to provide access to DFHDPWB and DFHADWB1:

```
For DFHDPWB:

URIMAP:       ADPM                - URIMAP name
GROUP:        MYGROUP             - any suitable
USAGE:        SERVER              - For inbound requests
SCHEME:       HTTP                - Or use HTTPS for SSL
HOST:         *                   - * matches any host name
PATH:         /cicsapps/adpm/*    - Any path can be used, with final asterisk
TCPIPSERVICE:                     - If blank, applies to all ports
ANALYZER:     NO                  - Means do not run analyzer
CONVERTER:    CICS                - Means no converter used
TRANSACTION:  CWBA                - Default transaction ID
PROGRAM:      DFHDPWB             - App debugging profile manager

For DFHADWB1:

URIMAP:       ADWB1
GROUP:        MYGROUP
USAGE:        SERVER
SCHEME:       HTTP
HOST:         *
PATH:         /cicsapps/adpm/dfhadwb1
[Same path as for DFHDPWB, but with dfhadwb1 appended]
TCPIPSERVICE:
ANALYZER:     NO
CONVERTER:    CICS
TRANSACTION:  CWBA
PROGRAM:      DFHADWB1             - Graphics program
```

When URIMAP definitions are used, the path specified in the URL does not need to have any relationship to the resource; the linkage between the path and the resource is made by the URIMAP definition. For the URIMAP that points to DFHDPWB, you could specify a path that includes a page element, such as `/cicsapps/adpm/start.html`. The name of the page element does not matter. Note

that if you do this, when the Web browser makes a request for DFHADWB1, it omits the page element and appends `dfhadwb1` in its place. If the path in the first of the sample URIMAP definitions was `/cicsapps/adpm/start.html`, the path in the second sample URIMAP definition would still be `/cicsapps/adpm/dfhadwb1`.

The *CICS Internet Guide* explains how to set up URIMAP definitions.

The List profiles page

Use the “List profiles” page to display a list of debugging profiles:

- When you start the application debugging profile manager's Web interface, CICS displays the “List profiles” page.

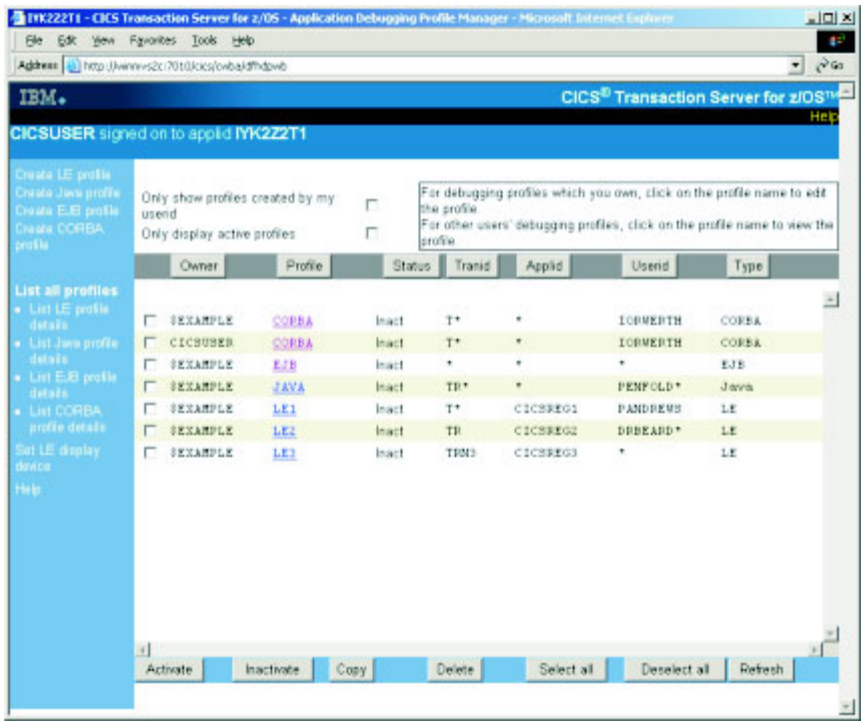


Figure 1. The List profiles page

When you use the debugging profile manager for the first time, CICS displays all profiles. Subsequently, CICS displays the profiles that were selected when you last used it.

If there are more profiles than can be displayed in the window, use the scrollbars to scroll backwards and forwards through the list. If you have no profiles, CICS displays an empty list.

There are four variants of the “List profiles” page:

List LE profiles
: Lists only the profiles for compiled language profiles

List Java profiles
: Lists only the profiles for Java programs

List EJB profiles
Lists only the profiles for enterprise beans

List CORBA profiles
Lists only the profiles for stateless CORBA objects

The behavior of these pages is identical to the “List profiles” page; however the information displayed on each is specific to the type of profile.

The list contains selected information from the debugging profile. The columns on the page are:

Owner
The userid of the profile owner; that is, of the user who created the profile.

Profile
The name of the profile

Status
The status of the profile (**Act** for Active, or **Inact** for Inactive)

The following columns display information specified when the profile is created:

Tranid Displays the contents of the **transaction** field

Program
On the “List profiles” and “List LE profiles” pages only , displays the contents of the **program** field.

Compile Unit
On the “List profiles” and “List LE profiles” pages only, displays the contents of **Compile Unit** field.

If the Compile Unit name is too long to display in the available space, the leading characters are displayed, followed by “**...**”. To display the Compile Unit name in full, click on the profile name.

Applid
Displays the contents of the **Applid** field

Userid
Displays the contents of the **Userid** field

Termid
On the “List profiles” and “List LE profiles” pages only, displays the contents of the **Terminal** field.

Type On the “List profiles” page only, displays the type of program specified in the debugging profile:
CORBA
CORBA object
EJB Enterprise bean
Java Java program
LE Compiled language program

Netname
On the “List LE profiles” page only, displays the contents of the **Netname** field.

Class On the “List Java profiles” and “List CORBA profiles” pages only, displays the contents of the **Class** field.

If the Class name is too long to display in the available space, the trailing characters are displayed, preceded by "...". To display the Class name in full, click on the profile name.

Bean On the "List EJB profiles" page only, displays the contents of the **Bean** field.

If the bean name is too long to display in the available space, the leading characters are displayed, followed by "..." To display the bean name in full, click on the profile name.

Method

On the "List EJB profiles" and "List CORBA profiles" pages only, displays the contents of the **Method** field.

If the Method name is too long to display in the available space, the leading characters are displayed, followed by "...". To display the Method name in full, click on the profile name.

You can change the way information is displayed on the "List profiles" page:

Selecting which profiles are displayed

Use the checkboxes at the top of the page to select which debugging profiles are displayed. The options are:

- Display all profiles
- Display all profiles that you created
- Display all active profiles
- Display only active profiles that you created

Sorting the list

Use the buttons above each column to re-display the list in the sequence determined by the contents of the column. For example, to re-display the profiles in sequence of program name, click the **Program** button. CICS uses the EBCDIC sorting sequence when it re-displays the list.

Your choice of which profiles are displayed, and your chosen sequence, are preserved, and used the next time you use the debugging profile manager.

Buttons on the List profiles page

The buttons at the head of the following columns are used to re-display the list of profiles in sequence:

Owner
Profile
Status
Tranid
Program
Compile Unit
Applid
Userid
Termid
Netname
Type

The following buttons are inactive, and cannot be used to re-sequence the list of profiles:

Class

Bean

Method

Other actions are performed using the buttons at the bottom of the “List profiles” page:

Activate
Activate selected profiles. See “Activating debugging profiles with the Web interface” on page 23 for more information.

Inactivate
Inactivate selected profiles. See “Inactivating debugging profiles with the Web interface” on page 23 for more information.

Copy
Copy selected profiles. See “Copying debugging profiles with the Web interface” on page 24 for more information.

Delete
Delete selected profiles. See “Deleting debugging profiles with the Web interface” on page 25 for more information.

Select all
Selects all the profiles in the list.

Deselect all
Deselects all the profiles in the list

Refresh
Refresh the “List profiles” page. The list is updated to show any changes that you, and other users, have made.

Creating a debugging profile with the Web interface

You can create debugging profiles in these ways:
- You can create a completely new profile by entering all the information needed to define the profile
- You can base the new profile on an existing profile

Creating a new profile

Starting with the “List all profiles” page, follow these steps:
- To create a profile for a compiled language (Language Environment®) program
  1. Click on **Create compiled profile**. CICS displays the “Create compiled profile” page.
  2. Complete the information that you need to specify your profile. See “The Create compiled profile page” on page 25 for details.
  3. Click on the **Create** button. CICS checks that you have entered valid data.
     - If your data is valid, the profile is saved
     - If your data contains an error, a message is displayed. Re-enter the data, and click on the **Create** button again.
  4. Click on **List all profiles** to return to the “List all profiles” page.
- To create a profile for a Java program
  1. Click on **Create Java profile**. CICS displays the “Create Java profile” page.
  2. Complete the fields that you need to specify your profile.
  3. Click on the **Create** button. CICS checks that you have entered valid data.
     - If your data is valid, the profile is saved

– If your data contains an error, a message is displayed. Re-enter the data, and click on the **Create** button again.

4. Click on **List all profiles** to return to the “List profiles” page.

Basing a new profile on an existing profile

You can create a new debugging profile using an existing profile as a starting point. The steps you take depend upon whether the original profile is owned by you or another user:

If you own the profile

Follow the steps described in “Changing a debugging profile with the Web interface” on page 24. Before you save the profile, give it a new name.

If another user owns the profile:

Follow the steps below.

1. Copy the profile, following the steps described in “Copying debugging profiles with the Web interface” on page 24.
2. Make any changes to the profile by following the steps described in “Changing a debugging profile with the Web interface” on page 24.

Activating debugging profiles with the Web interface

To activate debugging profiles, start with the “List all profiles” page, and follow these steps:

1. Use the checkboxes at the top of the page to ensure that the display includes the profiles you want to activate.
2. Scroll the list to a profile that you want to activate.
3. Select the profile using the checkbox to the left of the profile name.
4. Repeat steps 2 through 3 to select all the profiles you want to activate.
5. Click the **Activate** button. By default, if any of the selected profiles is for a compiled language (Language Environment) program, CICS displays the “Set compiled display device” page.

   If none of the selected profiles is for a compiled language programs, CICS refreshes the “List all profiles” page.

   You can choose not to see the “Set compiled display device” page when you activate profiles. See “Setting the display device” on page 34 for more information.

Note: If you change a profile while it is active, the changes take effect immediately: the next time a program is started, the changed parameters are used to decide if the program should run under the debugger's control.

Inactivating debugging profiles with the Web interface

To inactivate debugging profiles, start with the “List profiles” page, and follow these steps:

1. Use the checkboxes at the top of the page to ensure that the display includes the profiles you want to inactivate.
2. Scroll the list to a profile that you want to inactivate.
3. Select the profile using the checkbox to the left of the profile name.
4. Repeat steps 2 through 3 to select all the profiles you want to inactivate.

5. Click the **Inactivate** button. The “List profiles” page is refreshed

Viewing a debugging profile with the Web interface

If you are not the owner of a debugging profile, you can view its contents, but you cannot change it. To view the contents of a debugging profile, start with the “List all profiles” page, and follow these steps:

1. Use the checkboxes at the top of the page to ensure that the profile you want to change is displayed.
2. Scroll the list to the profile you want to change.
3. Click on the profile name. CICS displays the “View compiled profile” page, or the “View Java profile” page.
4. Click on **List all profiles** to return to the “List all profiles” page.

Note: If you follow these steps for a profile that you own, CICS displays the “Edit compiled profile” page, or the “Edit Java profile” page. In this case, you will be able to modify the contents of the profile.

Changing a debugging profile with the Web interface

If you are the owner of a debugging profile, you can change its contents. Starting with the “List all profiles” page, follow these steps:

1. Use the checkboxes at the top of the page to ensure that the profile you want to change is displayed.
2. Scroll the list to the profile you want to change.
3. Click on the profile name. CICS displays the “Create compiled profile” page, or the “Create Java profile” page.
4. Make your changes to the displayed fields.
5. Click the **Replace** button.
6. Click on **List all profiles** to return to the “List all profiles” page.

Note:

1. If you change the name of the profile, the debugging profile manager creates a new profile with the new name, and leaves the original profile unchanged.
2. If you follow these steps for a profile that you do not own, CICS displays the “View compiled profile” page, or the “View Java profile” page. In this case, you will not be able to modify the contents of the profile.
3. If you change a profile while it is active, the changes take effect immediately: the next time a program is started, the changed parameters are used to decide if the program should run under the debugger's control.

Copying debugging profiles with the Web interface

You can copy profiles that are owned by other users to create identical profiles that you own. Each new profile has the same name as the one that was copied. Starting with the “List profiles” page, follow these steps:

1. Use the checkboxes at the top of the page to ensure that the display includes the profiles you want to copy.

2. Scroll the list to a profile that you want to copy.
3. Select the profile using the checkbox to the left of the profile name.
4. Repeat steps 2 through 3 to select all the profiles you want to copy.
5. Click the **Copy** button. The profiles are copied, and the “List profiles” page is refreshed.

Note: You cannot copy a profile that you own. If you want to create a new profile based on a profile that you own, follow the steps described in “Changing a debugging profile with the Web interface” on page 24. Before saving the changed profile, give it a new name.

Deleting debugging profiles with the Web interface

To delete debugging profiles, start with the “List profiles” page, and follow the steps below. You cannot delete a profile that is owned by another user.

1. Use the checkboxes at the top of the page to ensure that the display includes the profiles you want to delete.
2. Scroll the list to a profile you want to delete.
3. Select the profile using the checkbox to the left of the profile name.
4. Repeat steps 2 through 3 to select all the profiles you want to delete.
5. Click the **Delete** button. The profiles are deleted, and the “List profiles” page is refreshed.

Deleting the sample profiles

Although, in general, you cannot delete debugging profiles that are owned by another user, the sample profiles are handled as a special case, and you can delete them.

Be aware that, if you do delete the sample profiles:

- You may affect users who want to use the sample profiles.
- The only way to create them again is to have your system programmer re-initialize the debugging profiles data sets. However, if you do this, any other profiles which already exist will be deleted.

If you want to use the sample profiles, and you are concerned that other users may delete them, copy the samples. You will own the copies, and no-one else will be able to delete them.

If you don't want to see the sample profiles when you list profiles, display only the profiles that you own.

The Create compiled profile page

Use the “Create compiled profile” page to work with the contents of a debugging profile for compiled language (Language Environment) programs. You can use the page to perform the following functions:

Create a new profile

Initially, the page contains default values for some of the fields. You must supply other values to complete the profile.

Edit an existing profile
Initially, the page contains the values that were previously defined for the profile.

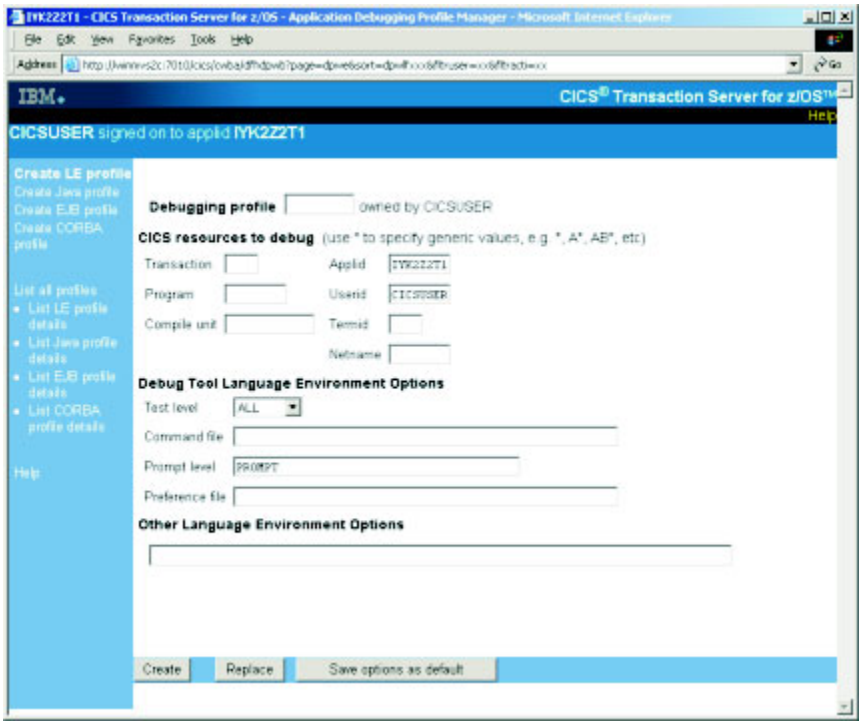


Figure 2. The Create compiled profile page

Fields on the Create compiled profile page

The fields on the “Create compiled profile” page are:

`Debugging profile`
Specifies the name of the profile.

If you are working with an existing profile, and you change the name that is displayed, the application debugging profile manager creates a new profile with the new name, and leaves the original profile unchanged.

Acceptable characters
`A-Z 0-9 $ @ #` Any lower case characters you enter are converted to upper case.

The following fields specify which programs should trigger the start of a debugging session when the profile is active.

`Transaction`
Specify a value in this field when you want to debug only those programs that run under the specified transaction id.

Acceptable characters

A-Z a-z 0-9 $ @ # . / - _ % & ¢ ? ! : | " = ¬ , ; < >

Because transaction ids are case-sensitive, any lower case characters you enter are **not** converted to upper case.

You can specify a generic value if you want to debug programs that run under a set of similarly-named transactions.

Note: Do not specify an alias transaction name in this field; CICS does not support the use of alias transaction names to select programs for debugging.

Program

Specify a value in this field when you want to debug only the specified program. In this context the program is the program as known to CICS, such as a load module name, initial program in a transaction or a program that has been XCTL'd or LINKed to.

Acceptable characters

A-Z 0-9 $ @ #

Any lower case characters you enter are converted to upper case.

You can specify a generic value if you want to debug a set of similarly-named programs.

Compile unit

Specify a value in this field when you want to debug only the specified compile unit. You can specify a generic value if you want to debug a set of similarly-named compile units. In this context the compile unit is the program as known to the compiler; for example, PROGRAM-ID for COBOL and the main PROCEDURE name for PL/I.

Acceptable characters

A-Z a-z 0-9 $ @ # . / - _ % & ¢ ? ! : | " = ¬ , ; < >

Because compile unit names are case-sensitive, any lower case characters you enter are **not** converted to upper case.

Applid

Specify a value in this field when you want to confine debugging to programs that run in the specified CICS region.

Acceptable characters

A-Z 0-9 $ @ #

Any lower case characters you enter are converted to upper case.

You can specify a generic value if you want to debug programs in a set of similarly-named regions.

The default value is the applid of the region where the application debugging profile manager is running, and is displayed at the top of the page.

Userid

Specify a value in this field when you want to confine debugging to programs that are being run by the specified user. The default value is the ID of the user that is using the debugging profile manager.

Acceptable characters

A-Z 0-9 $ @ #

Any lower case characters you enter are converted to upper case.

You can specify a generic value if you want to debug programs that are being run by a group of similarly-named users.

The default value is the ID of the user that is using the application debugging profile manager, and is displayed at the top of the page.

Important: The user ID specified here is not necessarily the owner of the profile: the owner of the profile is the user who created it.

Termid

Specify a value in this field when you want to confine debugging to programs that are being run at the specified terminal.

Acceptable characters

A-Z a-z 0-9 $ @ # . / - _ % & ¢ ? ! : | " = ¬ , ; < >

Restriction: You cannot specify a terminal ID that consists entirely of blanks

You can specify a generic value if you want to debug programs that are being run at a number of similarly-named terminals.

Important: The terminal specified here is not necessarily the terminal at which the debugging session is conducted. The terminal used for the debugging session is specified in the “Set compiled display device” page.

Netname

Specify a value in this field when you want to confine debugging to programs that are being run at terminals with the specified netname.

Acceptable characters

A-Z 0-9 $ @ #

Any lower case characters you enter are converted to upper case.

You can specify a generic value if you want to debug programs that are being run at a number of terminals with similar netnames.

The following fields specify options that are passed to Debug Tool. See *Debug Tool for z/OS and OS/390 User's Guide* for more information. You can save the values that you specify; the saved values are used by default each time you create a Language Environment debugging profile.

Test level

Specifies what conditions need to be met for Debug Tool to gain control for programs that match this profile. Select one of the following values:

ALL
ERROR
NONE

Command file

Specifies the primary commands file associated with the profile. You can specify the fully qualified name of a sequential data set or a member of a partitioned data set.

Prompt level
Specifies whether an initial commands list is unconditionally executed during program initialization. Enter one of the following:
PROMPT
NOPROMPT
command

Preference file
Specifies the preference file that Debug Tool uses when debugging programs that match this profile. You can specify the fully qualified name of a sequential data set or a member of a partitioned data set.

Language Environment options
Specifies Language Environment runtime options for programs that match this profile. When a program is selected for debugging because it matches the profile, the runtime options specified will override other runtime options that you may have in effect. For more information about defining runtime options for Language Environment, see the *CICS Application Programming Guide*.

Buttons on the Create compiled profile page

The buttons on the “Create compiled profile” page are:

Create
Create a new profile using the information entered on the page

Replace
Update an existing profile using the information entered on the page

Save options as default
Save the contents of the following fields. The saved values are used by default each time you create a Language Environment debugging profile.
Test level
Command file
Prompt
Preference file
Language Environment options

The Create Java profile page

Use the “Create Java profile” page to work with the contents of a debugging profile for Java programs. You can use the page to perform the following functions:

Create a new profile
Initially, the page contains default values for some of the fields. You must supply other values to complete the profile.

Edit an existing profile
Initially, the page contains the values that were previously defined for the profile.

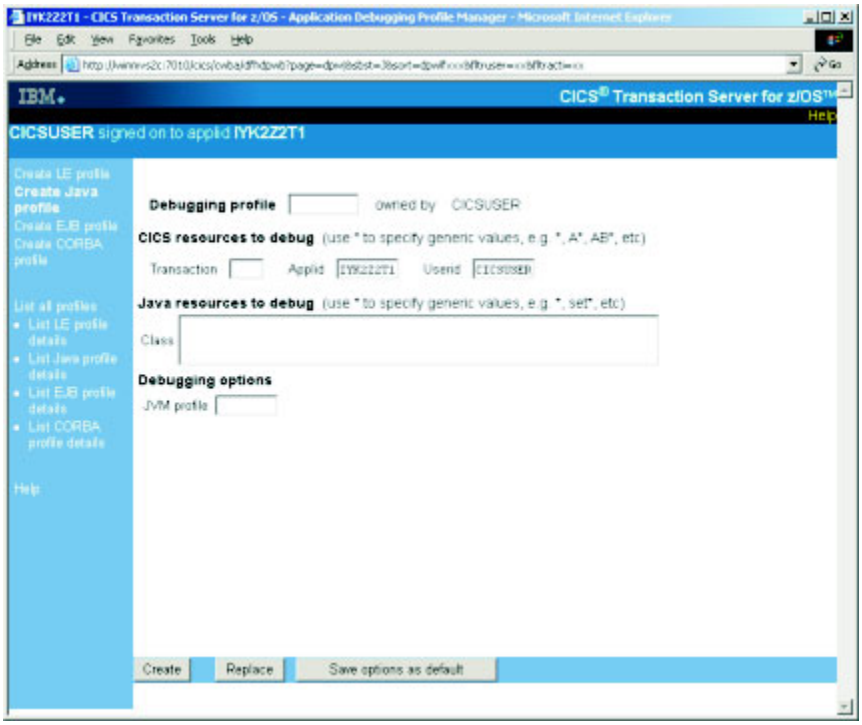


Figure 3. The Create Java profile page

Fields on the Create Java profile page

The fields on the “Create Java profile” page are:

Debugging Profile
Specifies the name of the profile.

If you are working with an existing profile, and you change the name that is displayed, the application debugging profile manager creates a new profile with the new name, and leaves the original profile unchanged.

Acceptable characters
A-Z 0-9 $ @ #
Any lower case characters you enter are converted to upper case.

The following fields specify which programs should trigger the start of a debugging session when the profile is active:

Transaction
Specify a value in this field when you want to debug only those programs that run under the specified transaction id.

Acceptable characters
A-Z a-z 0-9 $ @ # . / - _ % & ¢ ? ! : | " = ¬ , ; < >

You can specify a generic value if you want to debug programs that run under a set of similarly name transactions.

Note: Do not specify an alias transaction name in this field; CICS does not support the use of alias transaction names to select programs for debugging.

Applid
Specify a value in this field when you want to confine debugging to programs that run in the specified CICS region. The default value is the applid of the region where the Debugging Profile Manager is running.

Acceptable characters
A-Z 0-9 $ @ #
Any lower case characters you enter are converted to upper case.

You can specify a generic value if you want to debug programs in a set of similarly-named regions.

Userid
Specify a value in this field when you want to confine debugging to programs that are being run by the specified user. The default value is the ID of the user that is using the debugging profile manager.

Acceptable characters
A-Z 0-9 $ @ #
Any lower case characters you enter are converted to upper case.

You can specify a generic value if you want to debug programs that are being run by a group of similarly-named users.

Important: The user ID specified here is not necessarily the owner of the profile: the owner of the profile is the user who created it.

The following fields specify which Java resources should trigger the start of a debugging session when the profile is active:

Type
Specifies the type of Java resource that you want to debug:
Java Select this value when you want to debug a Java program.
EJB Select this value when you want to debug an enterprise bean.
CORBA
Select this value when you want to debug a stateless CORBA object.

Class
For Java programs and stateless CORBA objects only, specify a value in this field when you want to debug only the specified class. You can specify a generic value if you want to debug a set of similarly-named classes.

Bean
For enterprise beans only, specify a value in this field when you want to debug only the specified bean. You can specify a generic value if you want to debug a set of similarly-named beans.

Method
For enterprise beans and stateless CORBA objects only, specify a value in this field when you want to debug only the specified method.

When an inbound request initiated by a Java remote method invocation is received, the value specified is compared with the mangled name in the inbound request to determine if the profile matches the request. If it is possible that mangling can take place, do not specify a method name in the debugging

profile, but specify a generic method instead. For more information about mangling of method names, see *Java Applications in CICS*.

The following field specifies the debugging options for this profile. You can save the value that you specify; the saved value is used by default each time you create a Java debugging profile.

JVM profile
Specifies the name of the JVM profile that is used for Java programs that match this profile. The profile should specify that the Java program is to run in debug mode. You cannot specify a generic value for this parameter.

Buttons on the Create Java profile page

The buttons on the “Create Java profile” page are:

Create
Create a new profile using the information entered on the page

Replace
Update an existing profile using the information entered on the page

Save options as default
Save the contents of the following field. The saved value is used by default each time you create a Java debugging profile.
JVM profile

The View LE profile page

Use the “View LE profile” page to view the contents of a debugging profile for compiled language programs. You cannot change the profile with this page. The information displayed is described in “The Create compiled profile page” on page 25.

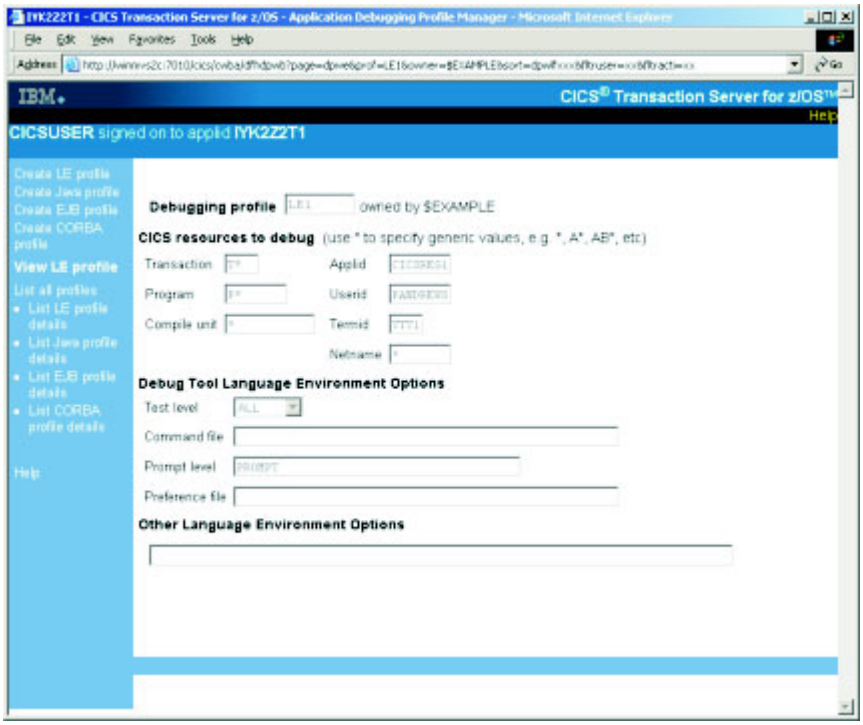


Figure 4. The View LE profile page

The View Java profile page

Use the “View Java profile” page to view the contents of a debugging profile for compiled language programs. You cannot change the profile with this page. The information displayed is described in “The Create Java profile page” on page 29.

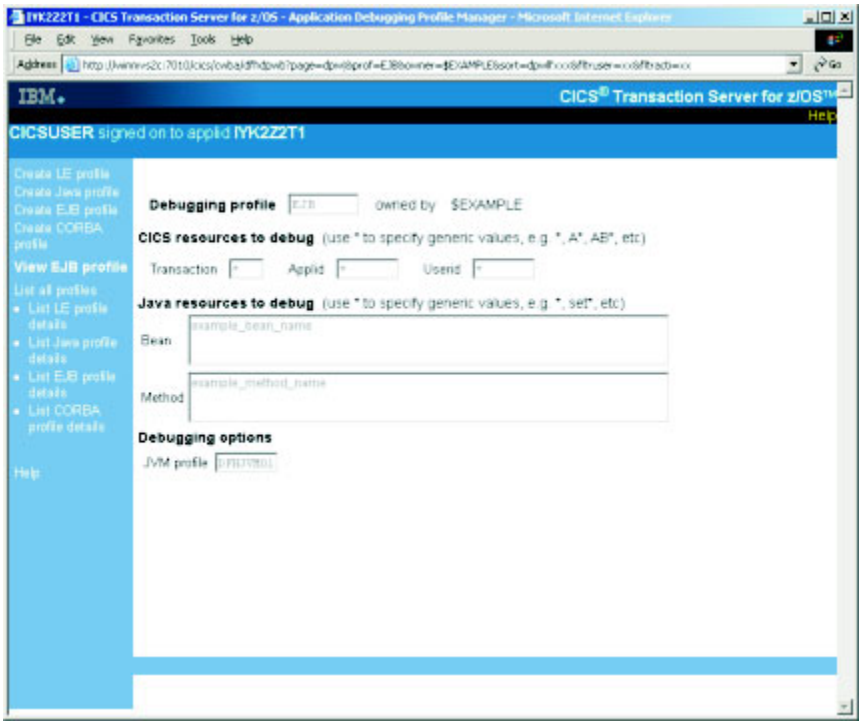


Figure 5. The View Java profile page

Setting the display device

When you have created a debugging profile for a compiled language (Language Environment) program, but before you can start debugging the application programs defined in the profile, you must specify the display device with which you will interact with the debugger. You can use:

a 3270 terminal
a debugging tool on a workstation.

The display device becomes associated with a debugging profile when you activate the profile, and remains associated with the profile until you make the profile inactive.

You can choose when you specify the display device:

- You can specify the display device *before* you make a profile active. The same display device will be associated with each profile that you subsequently make active.
- You can specify the display device *when* you make a profile active. The same display device will be associated with the profile that is made active. If you make more than one profile active at the same time (by selecting a number of profiles on the “List debugging profile” page) the same display device will be associated with them all.

Specifying the display device before you make a profile active

Starting with the “List profiles” page, follow these steps:

1. Click on Set LE display device. CICS displays the “Set LE display device” page.

2. Complete the details of the display device that you want to associate with the profile.
3. Select “In the future, do not show this page when activating profiles”.
4. Click the **Save and return** button. CICS saves the display device settings, and displays the “List profiles” page.

The “Set LE display device” page will not be displayed when you activate profiles; the settings you have supplied will be applied to all profiles for compiled language programs when you activate them.

Specifying the display device when you make a profile active

Starting with the “List profiles” page, follow these steps:

1. Click on Set LE display device. CICS displays the “Set LE display device” page.
2. Complete the details of the display device that you want to associate with the profile.
3. Using the checkbox, deselect “In the future, do not show this page when activating profiles”.
4. Click the **Save and return** button. CICS saves the display device settings, and displays the “List profiles” page.

CICS will display the “Set LE display device” page whenever you activate profiles for compiled language profiles. The page is initialized with the last settings you supplied. To change the settings:

1. Change any details of the display device that you want to associate with the set of profiles that is being activated.
2. Click the **Save and return** button. CICS saves the display device settings, and displays the “List profiles” page.

The Set compiled display device page

Use the “Set compiled display device” page to specify the display device with which you will interact with the debugger.

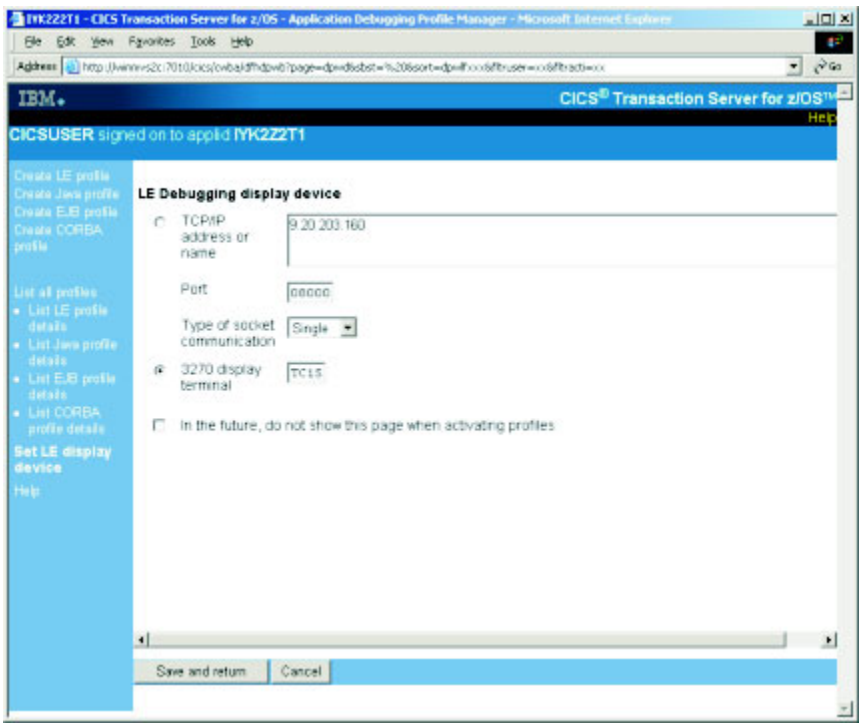


Figure 6. The Set LE display device page

Fields on the Set compiled display device page

The fields on the “Set compiled display device” page are:

Debugging display device
Use the radio buttons to select how you will interact with the debugger:

TCP/IP address or name
Specifies that you will interact with the debugger using a debugging client on your workstation. Supply the following information:

- The IP address or name of the host where the debugging client is running. By default, CICS inserts the IP address of the client which is running the browser, or — if there is a firewall between the browser and CICS — the IP address of the firewall.
- The port number at which the debugging client listens for a connection. Specify a value in the range 0 - 65535. The default is 8000.

Type of socket communication
For a debugging client on your workstation, specifies whether the debugging client and debugging server will communicate using a single socket or more than one socket.

Single
Use a single socket for communication. This is the default value, and is the preferred value when you use a WebSphere® Studio product as your debugging client.

Multiple
Use more than one socket for communication. You must specify this value when you use a VisualAge® product as your debugging client.

3270 display terminal
Specifies that you will interact with the debugger using a 3270 type terminal. Supply the following information:

- The terminal id of the terminal at which you will interact with the debugger.

  Important: The terminal specified here is not necessarily the terminal at which the transaction being debugged will run.

Buttons on the Set compiled display device page

The buttons on the "Set compiled display device" page are:

Save and return
Save the settings and return to the "List all profiles" page

Cancel
Return to the "List all profiles" page without saving the settings

Using the application debugging profile manager 3270 interface

Start the application debugging profile manager 3270 interface by running the CADP transaction:

1. On a 3270 screen, enter `CADP`.
2. Press `ENTER`. CICS displays the "List debugging profiles" screen.

The List debugging profiles screen

Use the "List debugging profiles" screen to display a list of debugging profiles. When you start the debugging profile manager, CICS displays the "List debugging profiles" screen.

```
CADP      -    CICS Application Debugging Profile Manager   -     IYK2Z2T1

List Debugging Profiles    (A=Activate,I=Inactivate,D=Delete,C=Copy)

  Owner    Profile  S Tran Program  Compile Unit Applid   Userid   Term Type
_ $EXAMPLE CORBA    I T*                         *        IORWERTH      Corb
_ CICSUSER CORBA    I T*                         *        IORWERTH      Corb
_ $EXAMPLE EJB      I *                          *        *             EJB
_ $EXAMPLE JAVA     I TR*                        *        PENFOLD*      Java
_ $EXAMPLE LE1      I T*   P*       *            CICSREG1 PANDREWS TTT1 LE
_ $EXAMPLE LE2      I TR   *        SAMPCOMPUN + CICSREG2 DRBEARD* TTT2 LE
_ $EXAMPLE LE3      I TRN3 PROG3    *            CICSREG3 *        TTT2 LE
_

7   profile(s). All profiles shown
Enter=Process PF1=Help 2=Filter 3=Exit 4=View 5=Create LE 6=Create Java
                9=Set display device 10=Edit 11=Sort
```

Figure 7. The List debugging profiles screen, showing the example profiles

When you use the debugging profile manager for the first time, CICS displays all the profiles that you own. Subsequently, CICS displays the profiles that were selected when you last used it.

If there are more profiles than can be displayed on one screen, use PF7 and PF8 to scroll backwards and forwards through the list. If you have no profiles, CICS displays an empty list.

The list contains selected information from the debugging profiles. The columns on the screen are:

Owner
The userid of the profile owner; that is, of the user who created the profile.

Profile
The name of the profile

S The status of the profile (**A** for Active, or **I** for Inactive)

The following columns display information specified when the profile is created:

Tran Displays the contents of the **transaction** field

Program
Displays the contents of the **program** field

Compile Unit
Displays the the first ten characters of the **Compile Unit** field. If the Compile Unit name is longer, a + to the right of the name shows that only part of the name is displayed.

Applid
Displays the contents of the **Applid** field

Userid
Displays the contents of the **Userid** field

Term Displays the contents of the **Terminal** field

Type Displays the type of program specified in the debugging profile:
Corb CORBA object
EJB Enterprise bean
Java Java program
LE Compiled language program

Not all the information in the debugging profile is displayed on the “List debugging profiles” screen. To display the additional information, move the cursor to the line that contains the profile, and press PF4.

You can change the way CICS displays information on the “List debugging profiles” screen:

Selecting which profiles are displayed
Use PF2 to cycle through the available options in turn. The options are
1. Display all the profiles in the system. This is the setting the first time you use the debugging profile manager
2. Display the profiles that you own
3. Display all active profiles

Sorting the list
Use PF11 to cycle through the available options in turn. The options are:
1. Re-display the profiles in sequence of profile name. This is the sequence the first time you use the debugging profile manager

2. Re-display the profiles in sequence of transaction ID
3. Re-display the profiles in sequence of program name
4. Re-display the profiles in sequence of owner

In each case, CICS uses the EBCDIC sorting sequence.

Your choice of which profiles are displayed, and your chosen sequence, are preserved, and used the next time you use the debugging profile manager.

Function keys for the List debugging profiles screen

The function keys for the “List debugging profiles” screen are:

PF1
Display the help screen

PF2
Selects which debugging profiles are displayed. This key cycles through the available options in turn.

PF3
End the debugging profile manager

PF4
Displays the “View LE debugging profile” screen or the “View Java debugging profile” screen for the profile on the line that contains the cursor.

PF5
Create a new debugging profile for a compiled language program

PF6
Create a new debugging profile for a Java program

PF7
Scroll backwards

PF8
Scroll forwards

PF9
Display the “Set LE debugging display device” screen

PF10
Edit the profile on the line that contains the cursor, using the “Create LE debugging profile” screen or the “Create Java debugging profile” screen.

PF11
Re-display the debugging profiles in a different sequence. This key cycles through the available options in turn.

Creating a debugging profile with the 3270 interface

You can create debugging profiles in these ways:

- You can create a completely new profile by entering all the information needed to define the profile.
- You can base the new profile on an existing profile.

Creating a new profile

Starting with the “List debugging profiles” screen, follow these steps:

1. Decide which type of profile you want to create.
   - To create a profile for a compiled language program, press PF5. CICS displays the “Create LE debugging profile” screen.
   - To create a profile for a Java program, press PF6. CICS displays the “Create Java debugging profile” screen.

2. Complete the fields that you need to specify your profile. You might need to use PF7 and PF8 to scroll the display.
3. Press ENTER. CICS checks that you have entered valid data.
   - If you have specified valid data, the profile is saved
   - If your data contains an error, CICS displays a message. Re-enter the data, and press ENTER again.
4. Press PF12 to return to the “List debugging profiles” screen.

Basing a new profile on an existing profile

You can create a new debugging profile using an existing profile as a starting point. The steps you take depend upon whether the original profile is owned by you or another user:

If you own the profile

Follow the steps described in “Changing a debugging profile with the 3270 interface” on page 41. Before you save the profile, give it a new name.

If another user owns the profile

Follow the steps below.

1. Copy the profile, following the steps described in “Copying debugging profiles with the 3270 interface” on page 41.
2. Make any changes to the profile by following the steps described in “Changing a debugging profile with the 3270 interface” on page 41.

Activating debugging profiles with the 3270 interface

To activate debugging profiles, start with the “List debugging profiles” screen, and follow these steps:

1. Use PF2 to ensure that the display includes the profiles you want to activate.
2. Use PF7 and PF8 to scroll to a profile that you want to activate.
3. Type **A** (for Activate) in the field to the left of the profile name.
4. Repeat steps 2 through 3 to select all the profiles you want to activate.
5. Press ENTER. By default, if any of the selected profiles is for a compiled language program, CICS displays the “Set LE debugging display device” screen.

   If none of the selected profiles is for a compiled language program, CICS refreshes the “List debugging profiles” screen.

   You can choose not to see the “Set LE debugging display device” screen when you activate profiles. See “Setting the display device” on page 53 for more information.

Note: If you change a profile while it is active, the changes take effect immediately: the next time a program is started, the changed parameters are used to decide if the program should run under the debugger's control.

Inactivating debugging profiles with the 3270 interface

To inactivate debugging profiles, start with the “List debugging profiles” screen, and follow these steps:

1. Use PF2 to ensure that the display includes the profiles you want to inactivate.
2. Use PF7 and PF8 to scroll to a profile that you want to inactivate.

3. Type **I** (for Inactivate) in the field to the left of the profile name.
4. Repeat steps 2 on page 40 through 3 to select all the profiles you want to inactivate.
5. Press ENTER. CICS makes the selected profiles inactive, and refreshes the "List debugging profiles" screen.

Viewing a debugging profile with the 3270 interface

To view the contents of a debugging profile, without changing it, start with the "List debugging profiles" screen, and follow these steps:

1. Use PF2 to ensure that the display includes the profile you want to view.
2. Use PF7 and PF8 to scroll to the profile you want to view.
3. Move the cursor to the line containing the profile that you want to view.
4. Press PF4. CICS displays the "View LE debugging profile" screen, or the "View Java debugging profile", depending on the options that were selected when the debugging profile was created.
5. When you have finished viewing the profile, press PF12 to return to the "List debugging profiles" screen.

Changing a debugging profile with the 3270 interface

If you are the owner of a debugging profile, you can change its contents. Starting with the "List debugging profiles" screen, follow these steps:

1. Use PF2 to ensure that the display includes the profile you want to change.
2. Use PF7 and PF8 to scroll to a profile you want to change.
3. Move the cursor to the line containing the profile that you want to change.
4. Press PF10. CICS displays "Create LE debugging profile" screen, or the "Create Java debugging profile" screen, depending on the options that were selected when the debugging profile was created.
5. Make your changes to the fields displayed on the screen.
6. Press PF10. CICS saves the changed profile.
7. Press PF12 to return to the "List debugging profiles" screen.

Note:

1. If you change the name of the profile, the debugging profile manager creates a new profile with the new name, and leaves the original profile unchanged.
2. If you change a profile while it is active, the changes take effect immediately: the next time a program is started, the changed parameters are used to decide if the program should run under the debugger's control.

Copying debugging profiles with the 3270 interface

You can copy profiles that are owned by other users, to create identical profiles that you own. Each new profile has the same name as the one that was copied. Starting with the "List debugging profiles" screen, follow these steps:

1. Use PF2 to ensure that the display includes the profiles you want to copy.
2. Use PF7 and PF8 to scroll to a profile that you want to copy.

3. Type **C** (for Copy) in the field to the left of the profile name.
4. Repeat steps 2 on page 41 through 3 to select all the profiles you want to copy.
5. Press ENTER. CICS copies the profiles, and refreshes the “List debugging profiles” screen.

Note: You cannot copy a profile that you own. If you want to create a new profile based on a profile that you own, follow the steps described in “Changing a debugging profile with the 3270 interface” on page 41. Before saving the changed profile, give it a new name.

Deleting debugging profiles with the 3270 interface

To delete debugging profiles, start with the “List debugging profiles” screen, and follow the steps below. You cannot delete a profile that is owned by another user.

1. Use PF2 to ensure that the display includes the profiles you want to delete.
2. Use PF7 and PF8 to scroll to a profile you want to delete.
3. You must make the profile inactive before you can delete it. To do this, see “Inactivating debugging profiles with the 3270 interface” on page 40.
4. Type **D** (for Delete) in the field to the left of the profile name.
5. Repeat steps 2 through 4 to select all the profiles you want to delete.
6. Press ENTER. CICS deletes the selected profiles, and refreshes the “List debugging profiles” screen.

Deleting the sample profiles

Although, in general, you cannot delete debugging profiles that are owned by another user, the sample profiles are handled as a special case, and you can delete them. Be aware that, if you do delete the sample profiles:

- You may affect users who want to use the sample profiles.
- The only way to create them again is to have your system programmer re-initialize the debugging profiles data sets. However, if you do this, any other profiles which already exist will be deleted.

If you want to use the sample profiles, and you are concerned that other users may delete them, copy the samples. You will own the copies, and no-one else will be able to delete them.

If you don't want to see the sample profiles when you list profiles, use PF2 on the “List debugging profiles” screen to display only the profiles that you own.

Combining actions on the List debugging profiles screen

From the “List debugging profiles screen”, you can activate, inactivate, delete and copy debugging profiles by typing the appropriate action character (**A**, **I**, **D** and **C** respectively) in the field to the left of the profile name. You can combine these actions on the “List debugging profiles” screen. For example, you can activate some profiles, and inactivate others in a single operation:

1. Use PF2 to ensure that the display includes the profiles you want to work with.
2. Use PF7 and PF8 to scroll to a profile you want to work with.
3. Type the action character in the field to the left of the profile name.
4. Repeat steps 2 and 3 to select all the profiles you want to work with.

5. Press ENTER. CICS processes the selected profiles, and refreshes the “List debugging profiles” screen. If any of the selected actions fail, processing stops, and the unprocessed action characters remain on the screen.

The Create LE debugging profile screen

Use the “Create LE debugging profile” screen to work with the contents of a debugging profile for compiled language programs. You can use the screen to perform the following functions:

Create a new profile
Initially, the screen contains default values for some of the fields. You must supply other values to complete the profile.

Edit an existing profile
Initially, the screen contains the values that were previously defined for the profile.

```
CADP      -    CICS Application Debugging Profile Manager   -     IYK2Z2T1

Create LE Debugging Profile ==>           for CICSUSER


CICS Resources To Debug (use * to specify generic values e.g. *, A*, AB*, etc.)
Transaction      ==>                                  Applid   ==> IYK2Z2T1
Program          ==>                                  Userid   ==> CICSUSER
Compile Unit     ==>                                  Termid   ==> TC15
                                                      Netname  ==> IYCWTC15

Debug Tool Language Environment Options
Test Level       ==> All                                      (All,Error,None)
Command File     ==>
Prompt Level     ==> PROMPT
Preference File  ==>

Other Language Environment Options
==>
==>
==>
==>

Enter=Create PF1=Help 2=Save options as defaults 3=Exit 10=Replace 12=Return
```

Figure 8. The CADP Create LE debugging profile screen

Fields on the Create LE debugging profile screen

The fields on the “Create LE debugging profile” screen are:

Create LE Debugging Profile
Specify the name of the profile. If you change the name of an existing profile, CADP creates a new profile with the new name, and leaves the original profile unchanged.

Acceptable characters
A-Z 0-9 $ @ #
Any lower case characters you enter are converted to upper case.

The following fields specify which programs should trigger the start of a debugging session when the profile is active.

Transaction
Specify a value in this field when you want to debug only those programs that run under the specified transaction id.

Acceptable characters
A-Z a-z 0-9 $ @ # . / - _ % & ¢ ? ! : | " = ¬ , ; < >
Because transaction ids are case-sensitive, any lower case characters you enter are **not** converted to upper case.

You can specify a generic value if you want to debug programs that run under a set of similarly-named transactions.

Note: Do not specify an alias transaction name in this field; CICS does not support the use of alias transaction names to select programs for debugging.

Program
Specify a value in this field when you want to debug only the specified program. In this context the program is the program as known to CICS, such as a load module name, initial program in a transaction or a program that has been XCTL'd or LINKed to.

Acceptable characters
A-Z 0-9 $ @ #
Any lower case characters you enter are converted to upper case.

You can specify a generic value if you want to debug a set of similarly-named programs.

Compile unit
Specify a value in this field when you want to debug only the specified compile unit. You can specify a generic value if you want to debug a set of similarly-named compile units. In this context the compile unit is the program as known to the compiler; for example, PROGRAM-ID for COBOL and the main PROCEDURE name for PL/I.

Acceptable characters
A-Z a-z 0-9 $ @ # . / - _ % & ¢ ? ! : | " = ¬ , ; < >
Because compile unit names are case-sensitive, any lower case characters you enter are **not** converted to upper case.

Applid
Specify a value in this field when you want to confine debugging to programs that run in the specified CICS region. The default value is the applid of the region where the Debugging Profile Manager is running.

Acceptable characters
A-Z 0-9 $ @ #
Any lower case characters you enter are converted to upper case.

You can specify a generic value if you want to debug programs in a set of similarly-named regions.

Userid
Specify a value in this field when you want to confine debugging to programs

that are being run by the specified user. The default value is the ID of the user who is using the debugging profile manager.

Acceptable characters
A-Z 0-9 $ @ #
Any lower case characters you enter are converted to upper case.

You can specify a generic value if you want to debug programs that are being run by a group of similarly-named users.

Important: The user ID specified here is not necessarily the owner of the profile: the owner of the profile is the user who created it.

Termid

Specify a value in this field when you want to confine debugging to programs that are being run at the specified terminal.

Acceptable characters
A-Z a-z 0-9 $ @ # . / - _ % & ¢ ? ! : | " = ¬ , ; < >
Restriction: You cannot specify a terminal ID that consists entirely of blanks

You can specify a generic value if you want to debug programs that are being run at a number of similarly-named terminals.

Important: The terminal specified here is not necessarily the terminal at which the debugging session is conducted. The terminal used for the debugging session is specified in the “Set LE debugging display device” screen.

Netname

Specify a value in this field when you want to confine debugging to programs that are being run at terminals with the specified netname.

Acceptable characters
A-Z 0-9 $ @ #
Any lower case characters you enter are converted to upper case.

You can specify a generic value if you want to debug programs that are being run at a number of terminals with similar netnames.

The following fields specify suboptions of the TEST runtime option, and are passed to Debug Tool. See the *Debug Tool for z/OS and OS/390 User's Guide* for more information. You can save the values that you specify; the saved values are used by default each time you create an LE debugging profile. For more information, see “Specifying default values for Debug Tool and LE options” on page 46.

Test level

Specifies which conditions raised by your program will cause Debug Tool to gain control. You can enter the following values:

All
Error
None

Command file

Specifies the primary commands file associated with the profile. You can specify the fully qualified name of a sequential data set or a member of a partitioned data set.

Prompt level
Specifies whether an initial commands list is unconditionally executed during program initialization. Enter one of the following:
PROMPT
NOPROMPT
command

Preference file
Specifies the preference file that Debug Tool uses when debugging programs that match this profile. You can specify the fully qualified name of a sequential data set or a member of a partitioned data set.

Other Language Environment options
Specifies Language Environment runtime options for programs that match this profile. When a program is selected for debugging because it matches the profile, the runtime options specified will override other runtime options that you may have in effect. For more information, see the *CICS Application Programming Guide*.

Function keys for the Create LE debugging profile screen

The function keys for the "Create LE debugging profile" screen are:

PF1
Display the help screen

PF2
Save the contents of the Debug Tool options, and the Language Environment options. See "Specifying default values for Debug Tool and LE options."

PF3
End the debugging profile manager

PF10
Update an existing profile with the information on the screen

PF12
Return to the "List debugging profiles" screen

Specifying default values for Debug Tool and LE options

You can specify default values for the following Debug Tool options, and the Language Environment options. The saved values are used by default each time you create a debugging profile for a compiled language program. The Debug Tool options are:

Test level
Command file
Prompt level
Preference file

To save the default values, start with the "Create LE debugging profile" screen, and follow these steps:

1. Type the default values that you want to specify for the Debug Tool options and Language Environment options
2. Press PF2. CICS saves the values that you have specified.

The values that you save will be used by default each time you create a new profile.

The Create Java debugging profile screen

Use the “Create Java debugging profile” screen to work with the contents of a debugging profile for Java programs. You can use the screen to perform the following functions:

Create a new profile
Initially, the screen contains default values for some of the fields. You must supply other values to complete the profile.

Edit an existing profile
Initially, the screen contains the values that were previously defined for the profile.

The screen is in two parts; use PF7 and PF8 to scroll between them.

```
CADP      -    CICS Application Debugging Profile Manager   -     IYK2Z2T1

Create Java Debugging Profile ==>            for CICSUSER

CICS Resources To Debug (use * to specify generic values e.g. *, A*, AB*, etc.)
Transaction      ==>                          Applid       ==> IYK2Z2T1
                                              Userid       ==> CICSUSER

Debugging Options
JVM Profile      ==>

Java Resources To Debug
Type             ==> J       (J=Java Applications, E=Enterprise Beans, C=Corba)

Class (Java Applications or Corba)
==>
==>
==>
==>

Press PF8 to set Bean and Method

Enter=Create PF1=Help 2=Save options as defaults 3=Exit 8=Forward
10=Replace 12=Return
```

```
CADP      -    CICS Application Debugging Profile Manager   -     IYK2Z2T1

Java Resources To Debug

Bean (Enterprise Beans only)
==>
==>
==>
==>

Method (Enterprise Beans or Corba)
==>
==>
==>
==>









Enter=Create PF1=Help 3=Exit 7=Back 10=Replace 12=Return
```

Figure 9. The CADP Create Java debugging profile screens

Fields on the Create Java debugging profile screen

The fields on the “Create Java debugging profile” screen are:

Create Java Debugging Profile
Specifies the name of the profile. If you change the name of an existing profile, CADP creates a new profile with the new name, and leaves the original profile unchanged.

Acceptable characters
A-Z 0-9 $ @ #
Any lower case characters you enter are converted to upper case.

The following fields specify which programs should trigger the start of a debugging session when the profile is active.

Transaction
Specify a value in this field when you want to debug only those programs that run under the specified transaction id.

Acceptable characters
A-Z a-z 0-9 $ @ # . / - _ % & ¢ ? ! : | " = ¬ , ; < >

You can specify a generic value if you want to debug programs that run under a set of similarly-named transactions.

Note: Do not specify an alias transaction name in this field; CICS does not support the use of alias transaction names to select programs for debugging.

Applid
Specify a value in this field when you want to confine debugging to programs that run in the specified CICS region. The default value is the applid of the region where the Debugging Profile Manager is running.

Acceptable characters
A-Z 0-9 $ @ # Any lower case characters you enter are converted to upper case.

You can specify a generic value if you want to debug programs in a set of similarly-named regions.

Userid
Specify a value in this field when you want to confine debugging to programs that are being run by the specified user. The default value is the ID of the user that is using the debugging profile manager.

Acceptable characters
A-Z 0-9 $ @ # Any lower case characters you enter are converted to upper case.

You can specify a generic value if you want to debug programs that are being run by a group of similarly-named users.

Important: The user ID specified here is not necessarily the owner of the profile: the owner of the profile is the user who created it.

The following field specifies the debugging options for this profile. You can save the value that you specify; the saved value is used by default each time you create a Java debugging profile. To save the value, press PF2.

JVM profile
Specifies the name of the JVM profile that is used for Java programs that match this profile. The profile should specify that the Java program is to run in debug mode. For more information about the JVM profile, see Setting up JVM profiles and JVM properties files *Java Applications in CICS*.

The following fields specify which Java resources should trigger the start of a debugging session when the profile is active:

Type
Specifies the type of Java resource that you want to debug:
J Enter this value when you want to debug a Java program.
E Enter this value when you want to debug an enterprise bean.
C Enter this value when you want to debug a stateless CORBA object.

Class
For Java programs and stateless CORBA objects only, specify a value in this field when you want to debug only the specified class. You can specify a generic value if you want to debug a set of similarly-named classes.

Bean
For enterprise beans only, specify a value in this field when you want to debug only the specified bean. You can specify a generic value if you want to debug a set of similarly-named beans.

Method
For enterprise beans and CORBA objects only, specify a value in this field when you want to debug only the specified method.

When an inbound request initiated by a Java remote method invocation is received, the value specified is compared with the mangled name in the inbound request to determine if the profile matches the request. If it is possible that mangling can take place, do not specify a method name in the debugging profile, but specify a generic method instead. For more information about mangling of method names, see *Java Applications in CICS*.

Function keys for the Create Java debugging profile screen

The function keys for the “Create Java debugging profile” screen are:
PF1
Display the help screen
PF2
Save the contents of the **JVM profile** field. The saved value is used by default each time you create a Java debugging profile.
PF3
End the debugging profile manager
PF7
Scroll forward
PF8
Scroll backward
PF10
Update an existing profile with the information on the screen
PF12
Return to the “List debugging profiles” screen

The View LE debugging profile screen

Use the “View LE debugging profiles” screen to view the contents of a debugging profile for compiled language programs. You cannot change the profile with this screen.

```
CADP      -    CICS Application Debugging Profile Manager   -     IYK2Z2T1

View LE Debugging Profile LE1      for $EXAMPLE


CICS Resources To Debug
Transaction      ==> T*                              Applid   ==> CICSREG1
Program          ==> P*                              Userid   ==> PANDREWS
Comp Unit        ==> *                               Termid   ==> TTT1
                                                     Netname  ==> *

Debug Tool Language Environment Options
Test Level       ==> All                                    (All,Error,None)
Command File     ==>
Prompt Level     ==> PROMPT
Preference File  ==>

Other Language Environment Options
==>
==>
==>
==>

PF1=Help 3=Exit 12=Return
```

Figure 10. The View LE debugging profile screen, showing example profile LE1

The information displayed is described in “The Create LE debugging profile screen” on page 43.

Function keys for the View LE debugging profile screen

The function keys for the “View LE debugging profile” screen are:

PF1
: Display the help screen

PF3
: End the debugging profile manager

PF12
: Return to the “List debugging profiles” screen

The View Java debugging profile screen

Use the “View Java debugging profiles” screen to view the contents of a debugging profile for Java programs. You cannot change the profile with this screen.

The screen is in two parts; use PF7 and PF8 to scroll between them.

```
CADP      -    CICS Application Debugging Profile Manager   -     IYK2Z2T1

View Java Debugging Profile EJB      for $EXAMPLE

CICS Resources To Debug
Transaction      ==> *                        Applid       ==> *
                                              Userid       ==> *

Debugging Options
JVM Profile      ==> DFHJVM01

Java Resources To Debug
Type             ==> E       (J=Java Applications, E=Enterprise Beans, C=Corba)

Class (Java Applications or Corba)
==>
==>
==>
==>




PF1=Help 3=Exit 8=Forward 12=Return
```

```
CADP      -    CICS Application Debugging Profile Manager   -     IYK2Z2T1

View Java Debugging Profile EJB      for $EXAMPLE

Bean (Enterprise Beans only)
==> example_bean_name
==>
==>
==>

Method (Enterprise Beans or Corba)
==> example_method_name
==>
==>
==>









PF1=Help 3=Exit 7=Back 12=Return
```

Figure 11. The View Java debugging profile screen, showing example profile EJB

The information displayed is described in “The Create Java debugging profile screen” on page 47.

Function keys for the View Java debugging profile screen

The function keys for the “View Java debugging profile” screen are:

PF1
Display the help screen

PF3
End the debugging profile manager

PF7
Scroll forward

PF8
Scroll backward

PF12
Return to the “List debugging profiles” screen

Setting the display device

When you have created a debugging profile for a compiled language program, but before you can start debugging the application programs defined in the profile, you must specify the display device with which you will interact with the debugger. You can use:

a 3270 terminal
a debugging tool on a workstation.

The display device becomes associated with a debugging profile when you activate the profile, and remains associated with the profile until you make the profile inactive.

You can choose when you specify the display device:

- You can specify the display device before you make a profile active. The same display device will be associated with each profile that you subsequently make active.
- You can specify the display device when you make a profile active. The same display device will be associated with the profile that is made active. If you make more than one profile active at the same time (by selecting a number of profiles on the “List debugging profile” screen) the same display device will be associated with them all.

Specifying the display device before you make a profile active

Starting with the “List debugging profiles” screen, follow these steps:

1. Press PF9 to display the “Set LE debugging display device” screen.
2. Complete the details of the display device that you want to associate with the profile.
3. Type “No” in the “Display this panel on each LE profile activation” field.
4. Press ENTER. CICS saves the display device settings, and displays the “List debugging profiles” screen.

The “Set LE debugging display device” screen will not be displayed when you activate profiles; the settings you have supplied will be applied to all compiled language profiles when you activate them.

Specifying the display device when you make a profile active

Starting with the “List debugging profiles” screen, follow these steps:

1. Press PF9 to display the “Set LE debugging display device” screen.
2. Complete the details of the display device that you want to associate with the profile.
3. Type “Yes” (the default value) in the “Display this panel on each LE profile activation” field.
4. Press ENTER. CICS saves the display device settings, and displays the “List debugging profiles” screen.

CICS will display the “Set LE debugging display device” screen whenever you activate profiles for compiled language programs. The screen is initialized with the last settings you supplied. To change the settings:

1. Change any details of the display device that you want to associate with the set of profiles that is being activated.
2. Press ENTER. The display device settings are saved, and associated with the profile. CICS displays the “List debugging profiles” screen.

The Set LE debugging display device screen

Use the “Set LE debugging display device” screen to specify the display device with which you will interact with the debugger.

```
CADP      -    CICS Application Debugging Profile Manager   -     IYK2Z2T1

Set LE Debugging Display Device

Debugging Display Device
Session Type            ==>  3270                        (3270,TCP)
3270 Display Terminal   ==>  TC15

TCP/IP Name Or Address
==>
==>
==>
==>
Port                    ==>  08000

Type of socket communication ==> Single                  (Single,Multiple)


Display this panel on LE profile activation ==> YES




Enter=Save and return PF1=Help 3=Exit 12=Cancel
```

Figure 12. The CADP Set LE debugging display device screen

Fields on the Set LE debugging display device screen

The fields on the “Set LE debugging display device” screen are:

Session type
Specifies how you will interact with Debug Tool:

3270
You will interact with Debug Tool using a 3270 type terminal. Specify the terminal id in the display id field. This is the default value.

TCP
You will interact with Debug Tool using a debugging client on your workstation. The client will communicate with Debug Tool using TCP/IP. Specify the port number at which the client listens for a connection in the port field.

3270 display terminal
When the session type is 3270, specify the terminal id of the terminal with which you will interact with Debug Tool. The default value is the id of the terminal at which you running CADP.

Important: The terminal specified here is not necessarily the terminal at which the transaction being debugged will run.

TCP/IP name or address
When the session type is TCP, specify the IP address or name of the host where the debugging client is running.

Port
When the session type is TCP, specify the port number at which the debugging client listens for a connection. Specify a value in the range 0 - 65535. The default is 8000.

Type of socket communication
When the session type is TCP, specifies whether the debugging client and debugging server will communicate using a single socket or more than one socket.

Single
Use a single socket for communication. This is the default value, and is the preferred value when you use a WebSphere Studio product as your debugging client.

Multiple
Use more than one socket for communication. You must specify this value when you use a VisualAge product as your debugging client.

Display this panel on LE profile activation
Specifies whether you want to display the “Set LE debugging display device” screen whenever you activate debugging profiles for compiled language programs:

YES Display the “Set LE debugging display device” screen whenever debugging profiles are activated. This is the default behavior.

NO Do not display the “Set LE debugging display device” screen whenever debugging profiles are activated. The display device that you specify will be associated with all the profiles that you activate.

Function keys for the Set LE debugging display device screen

The function keys for the “Set LE debugging display device” screen are:

PF1
Display the help screen

PF3
End the debugging profile manager

PF12
Return to the “List debugging profiles” screen

Chapter 3. CBAM - BTS browser

Use CBAM to browse the CICS business transaction services (BTS) objects (process-types, processes, activities, containers, events and timers) known to this region.

CBAM is described in the *CICS Business Transaction Services* manual.

© Copyright IBM Corp. 1982, 2011

Chapter 4. CCRL - certificate revocation lists transaction

Use the CCRL transaction to create and update the certificate revocation lists (CRLs) that are stored in an LDAP server. You only need to use CCRL if you are implementing SSL in your CICS regions and want each connection checked for a revoked certificate during the SSL handshake.

The CCRL transaction specifies the location of CRL repositories on the World Wide Web. CICS downloads the lists from the CRL repository at the specified URL and stores it in the LDAP server. You can specify more than one URL if you need to access multiple CRL repositories from different certificate authorities.

Before you run the CCRL transaction, you must have the following set up in CICS:

- An LDAP server that is set up and configured to store the certificate revocation lists. See the *CICS RACF Security Guide* for more information.
- The **CRLPROFILE** system initialization parameter defined with the name of the profile that authorizes CICS to access the LDAP server.

You can run the CCRL transaction from a terminal or from a START command. If you want to schedule regular updates, use the START command option.

To run the transaction from a terminal, ensure that the terminal accepts mixed case so that you can enter the URLs.

1. Enter the following command: `CCRL` *`url-list`* where *url-list* is a space-delimited list of URLs that contain the certificate revocation lists that you want to download. For example, you could specify:

   ```
   CCRL  http://crl.CertificateAuthority.com/CRLList1.crl
       http://crl.CertificateAuthority.com/CRLList2.crl
   ```

2. Enter the administrator distinguished name and password for the LDAP server. This allows CICS to update the LDAP server with the CRLs that it downloads. The name and password are set in the file `slapd.conf` in the LDAP server.

To run the transaction from a START command, using the following syntax: `EXEC CICS START TRANSID(CCRL) FROM (admin://`*`adminDN:adminPW url-list`*`) LENGTH (`*`url-list-length`*`) [INTERVAL(`*`hhmmss`*`)|TIME(`*`hhmmss`*`)]`, where *url-list* is a space-delimited list of URLs that contain the certificate revocation lists that you want to download, *url-list-length* is the length of the URL list, and *hhmmss* is the interval or expiration time at which the CCRL transaction is scheduled to run. For example you could specify:

```
EXEC CICS START TRANSID(CCRL)
          FROM ('admin://cn=ldapadmin:cics31ldap
       http://crl.CertificateAuthority.com/CRLList1.crl
       http://crl.CertificateAuthority.com/CRLList2.crl')
        LENGTH(124) INTERVAL(960000)
```

This example schedules the CCRL transaction to run in 96 hours. If you enter an invalid URL, you receive an error message.

© Copyright IBM Corp. 1982, 2011

Chapter 5. CDBC - database control menu

Use CDBC to connect CICS to DBCTL, or to disconnect CICS from DBCTL. CDBC is applicable only to CICS systems using the IMS/ESA Database Control (DBCTL) interface.

For detailed guidance on using the CDBC transaction with databases controlled by DBCTL, see the *CICS IMS Database Control Guide*.

You can start CDBC from any terminal, from a card reader/line printer (CRLP) or DASD sequential terminal, or from the operating system console.

There are two ways of commencing this transaction:

- Type the identifier, CDBC, on the command line of your display, followed by the required keyword, which can be either:

  ```
  CDBC CONnect [SUFfix(startup)]
  ```

  where **startup** is a two-digit startup table suffix, or:

  ```
  CDBC DISconnect [IMMediate]
  ```

- Type the identifier, CDBC, on the command line of your display, and press the ENTER key. You get the following display:

```
CDBC               CICS-DBCTL CONNECTION/DISCONNECTION               93.259
                                                                    13:33:20

          Select one of the following:

               1 Connection
               2 ORDERLY disconnection
               3 IMMEDIATE disconnection


          Option Selection     ==>
          Startup Table Suffix ==> 00
          DBCTL ID Override    ==>



  Status of the Interface: DFHDB8290I DBCTL not connected to CICS.
              CICS APPLID: IYAHZCD2
                 DBCTL ID:
     Startup Table Suffix:


PF1 = Help   2 = Refresh   3 = End
```

Figure 13. CDBC transaction: initial screen

As you can see from this display, you can select a CICS-DBCTL connection, an orderly disconnection, or an immediate disconnection.

This chapter describes:

- “Connecting CICS to DBCTL” on page 62
- “Disconnecting CICS from DBCTL” on page 62
- “Using the HELP panel” on page 63

© Copyright IBM Corp. 1982, 2011

Connecting CICS to DBCTL

You can connect CICS to DBCTL by either:

- Typing CDBC CONnect [SUFFIX(xx)]
- Typing '1' in the Option Selection field indicated by ==>. You can also specify a startup table suffix in the Startup Table Suffix field. If you leave this field blank you get the existing suffix, if one exists.

If you want to proceed with the CICS-DBCTL connection, press PF5 as requested on the display. The "status" line of the display is then updated. (You can find out the current status of the connection at any time by pressing PF2, which refreshes the display.)

You should get the following message:

```
DFHDB8207D DBCTL Connection requested. Press PF5
    to confirm.
```

This display also shows you the application identifier of the CICS system (in this case, DBDCCICS), and the identifier of the DBCTL system to which CICS is connected. The DBCTL identifier is available only when the connection has been made. You can also see the suffix of the startup table for this connection.

The following display shows you the effect of asking for option 1:

```
CDBC               CICS-DBCTL CONNECTION/DISCONNECTION                93.259
                                                                     13:33:59

          Select one of the following:

               1 Connection
               2 ORDERLY disconnection
               3 IMMEDIATE disconnection


          Option Selection     ==> 1
          Startup Table Suffix ==> 00
          DBCTL ID Override    ==>

  DFHDB8207D Connection to DBCTL requested. Press PF5 to confirm.

  Status of the Interface: DFHDB8290I DBCTL not connected to CICS.
              CICS APPLID: IYAHZCD2
                 DBCTL ID:
     Startup Table Suffix:


PF1 = Help   2 = Refresh   3 = End
```

Figure 14. CDBC transaction: option 1 screen

Disconnecting CICS from DBCTL

To disconnect CICS from DBCTL, you can either:

- Type

  ```
  CDBC DISConnect [IMMediate]
  ```

- Choose either option 2 or option 3 of the CDBC panel.

Option 2 gives you an orderly disconnection. An orderly disconnection allows all existing CICS-DBCTL tasks to be completed before CICS is disconnected from DBCTL.

The following display shows you the effect of asking for option 2:

```
CDBC               CICS-DBCTL CONNECTION/DISCONNECTION                93.259
                                                                     13:39:20

          Select one of the following:

               1 Connection
               2 ORDERLY disconnection
               3 IMMEDIATE disconnection


          Option Selection     ==> 2
          Startup Table Suffix ==> 00
          DBCTL ID Override    ==>

  DFHDB8209D DBCTL orderly disconnection requested. Press PF5 to confirm.

  Status of the Interface: DFHDB8293I DBCTL connected and ready.
              CICS APPLID: IYAHZCD2
                 DBCTL ID: SYS2
     Startup Table Suffix: 00


PF1 = Help   2 = Refresh   3 = End
```

Figure 15. CDBC transaction: option 2 screen

Option 3 gives you an immediate disconnection. This allows only current DL/I requests to DBCTL from this CICS system to be completed before CICS is disconnected from DBCTL.

Specifying a DBCTL system identifier (DBCTLID)

A field on the CDBC panel enables you to use the CDBC transaction to specify a DBCTLID to override the one in the DRA startup table. The syntax for connection to DBCTL using CDBC is:

```
CDBC CONNECT [SUFFIX(xx)] [DBCTLID(yyyy)]
```

For more details on using CDBC, see the *CICS IMS Database Control Guide*.

Using the HELP panel

If you press the help key (PF1), the following panel is displayed:

```
               HELP : CICS-DBCTL CONNECTION/DISCONNECTION

To CONNECT to DBCTL, select option 1. You can also specify a startup
 table suffix, or accept the existing suffix. The id of the DBCTL system is
 obtained from the startup table, but can be optionally overridden.

To DISCONNECT from DBCTL, select option 2 or option 3.

   Select option 2 for ORDERLY disconnection: this allows all CICS-DBCTL
   transactions from this CICS to complete before disconnecting from DBCTL.

   Select option 3 for IMMEDIATE disconnection: this allows all CICS-DBCTL
   requests from this CICS to complete before disconnecting from DBCTL.
--------------------------------------------------------------------------------
Displayed information (press PF2 to refresh the information):
   STATUS OF THE INTERFACE The current status of the connection to DBCTL.
   CICS APPLID             The application identifier for this CICS system.

   Displayed when available:
   DBCTL ID                 Identifier of the DBCTL system with which this
                            CICS system is communicating.
   STARTUP TABLE SUFFIX     Suffix used when CICS was connected to DBCTL.

                PRESS ENTER TO RETURN TO SELECTION SCREEN
```

Figure 16. CDBC transaction: the HELP panel

Chapter 6. CDBI - database control inquiry

Use CDBI to inquire about the status of the interface between CICS and DBCTL. You can start CDBI from 3270 (and 3270-like) terminals or from the operating system console.

For detailed information about the CICS-DBCTL interface and guidance on using the CDBI transaction with databases controlled by DBCTL, see the*CICS IMS Database Control Guide*.

Start the transaction by typing the identifier, CDBI, on the command line of your display, and press the ENTER key. You get the following display:
This display tells you the current status of the CICS-DBCTL connection, which can

```
CDBI                    CICS-DBCTL INTERFACE INQUIRY          91.184
                                                             11:23:50




            Status     : xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.
            CICS APPLID: DBDCCICS
            DBCTL ID   : SYS1









PF1 = Help   2 = Refresh   3 = End
```

Figure 17. CDBI transaction: initial screen

be one of the following:

```
DFHDB8290I DBCTL not connected to CICS.
DFHDB8291I DBCTL connect phase 1 in progress.
DFHDB8292I DBCTL connect phase 2 in progress.
DFHDB8293I DBCTL connected and ready.
DFHDB8294I DBCTL orderly disconnect in progress.
DFHDB8295I DBCTL immediate disconnect in progress.
DFHDB8296I DBCTL cannot be connected to CICS.
```

The display also shows you the application identifier of the CICS system (in this case, DBDCCICS), and the identifier of the DBCTL system to which CICS is connected.

Using the CDBI HELP panel

If you press the help key (PF1), you get the following display:

© Copyright IBM Corp. 1982, 2011

```
                  HELP : CICS-DBCTL INTERFACE INQUIRY

The CICS-DBCTL interface inquiry screen shows:

  STATUS OF THE INTERFACE    The status can be:

    DFHDB8290I DBCTL not connected to CICS.
    DFHDB8291I DBCTL connect phase 1 in progress.
    DFHDB8292I DBCTL connect phase 2 in progress.
    DFHDB8293I DBCTL connected and ready.
    DFHDB8294I DBCTL orderly disconnect in progress.
    DFHDB8295I DBCTL immediate disconnect in progress.
    DFHDB8296I DBCTL cannot be connected to CICS.

  CICS APPLID  The application identifier of this CICS system.

Displayed when available:

  DBCTL ID     The identifier of the DBCTL system with which this CICS
               is communicating

You can press PF2 to update (refresh) the information shown on screen.

                 PRESS ENTER TO RETURN TO INQUIRY SCREEN
```

Figure 18. CDBI transaction: the HELP panel

Chapter 7. CDBM - database control interface

Use the CDBM transaction to issue DBCTL operator commands from a CICS terminal. CDBM is applicable only to CICS systems using the IMS/ESA Database Control (DBCTL) interface.

CDBM also provides a means of maintaining a command file which stores commands. You may store commands for any reason, but repeated reuse is likely to be a frequent purpose. These stored commands may include more databases than the operator transaction panel has space for.

To use CDBM you must:

1. Have a DBCTL system running IMS/ESA 5.1 or later
2. DBCTL must have the **AOIS** parameter set to a value other than *N*. If values *A*, *C* or *R* are used, appropriate security controls must be used to allow the CICS-DBCTL interface to operate.
3. Generate and add to the DBCTL system, a PSB named DFHDBMP.

For detailed information about the CICS-DBCTL interface and guidance on using the CDBM transaction with databases controlled using DBCTL, see *CICS IMS Database Control Guide*.

Note: The CDBM transaction uses basic mapping support (BMS) and therefore this function is available only through those terminal devices that are supported by BMS.

This chapter describes:

- "Using CDBM Syntax"
- "Using DFHDBFK - The CDBM GROUP command file" on page 69
- "Using the CDBM HELP panels" on page 72

Using CDBM Syntax

The syntax of the CDBM transaction is:

```
CDBM /DBCTL operator command
```

where / is the default command recognition character (CRC) for DBCTL. In this case, the default CRC is the one that must be used.

Note:

1. IMS requires that each command be prefixed with the default CRC. The CRC is present only for syntax checking reasons; it does not determine the DBCTL to which the command is sent. You cannot use a CRC value to route a command to a particular DBCTL system through CDBM - it can only be sent to the one currently connected to CICS. This DBCTL may have its own CRC value which is different from the default one of /. This does not matter to CDBM however, because the / character is used only for syntax checking, and the command is presented to the connected DBCTL without a CRC, using the AIB interface
2. To ensure that the CDBM command succeeds, you might need to set the correct **AOIS** execution parameter. For example, AOIS=S ensures that the

© Copyright IBM Corp. 1982, 2011

command authorization is bypassed; with any other value, the command could fail the command authorization check.

- You can enter a DBCTL command, for example:

  ```
  CDBM /DISPLAY  DB  DEPT*
  ```

  or a group command, for example:

  ```
  CDBM /GROUP  SAMPLE  STO
  ```

  in this way.
- You can also start the transaction by typing the identifier, CDBM, on the command line of your display, as follows:

  ```
  CDBM
  ```

  Press the ENTER key. You get the panel shown in Figure 19.

```
CDBM                 CICS-DBCTL Operator Transaction              98.135
                                                                 13:24:20

Type IMS command.
             ____________________________________________________________
             ____________________________________________________________
             ____________________________________________________________
             ____________________________________________________________


For /DBDUMP or /DBRECOVER commands

Choose one. 1 1. Do not force end of volume
              2. Force end of volume


Press enter to display responses.


CICS APPLID DBDCCICS
DBCTL ID    SYS3


F1=Help  F2=Maintenance  F3=Exit  F5=Refresh  F12=Cancel
```

Figure 19. CICS-DBCTL operator transaction

On this panel you can enter a DBCTL command, for example:

```
/DISPLAY  DB  ALL
```

or a group command, for example:

```
/GROUP  SAMPLE  STA
```

An example of the use of a /GROUP command from the CICS-DBCTL Operator Transaction screen is shown in Figure 20 on page 69.

```
CDBM                CICS-DBCTL Operator Transaction              98.135
                                                                13:24:20

Type IMS command.
            /GROUP SAMPLE STA_____________________________________________
            _____________________________________________________________
            _____________________________________________________________
            _____________________________________________________________


For /DBDUMP or /DBRECOVER commands

Choose one. 1 1. Do not force end of volume
              2. Force end of volume


Press enter to display responses.


CICS APPLID DBDCCICS
DBCTL ID    SYS3


F1=Help  F2=Maintenance  F3=Exit  F5=Refresh  F12=Cancel
```

Figure 20. CICS-DBCTL operator transaction panel showing a GROUP command

Note: The group command *SAMPLE STA* referred to in this panel can be found in the “Using DFHDBFK - The CDBM GROUP command file.”

Responses to commands issued using CDBM are returned in a screen like the one shown in Figure 21, which shows the first of a number of screens issued in response to a /DISPLAY DB ALL command.

```
CDBM                     CICS-DBCTL IMS Responses             Screen 1
                                              Responses 1    to 18
                                                            More:   +
DATABASE  TYPE  TOTAL UNUSED  TOTAL UNUSED ACC  CONDITIONS
ACCUNTDB                                   UP   STOPPED, NOTOPEN, NOTINIT
ADMIDX1                                    UP   STOPPED, NOTOPEN, NOTINIT
ADMOBJ1                                    UP   STOPPED, NOTOPEN, NOTINIT
ADMOBJ2                                    UP   STOPPED, NOTOPEN, NOTINIT
ADMOBJ3                                    UP   STOPPED, NOTOPEN, NOTINIT
ADMSYSDF                                   UP   STOPPED, NOTOPEN, NOTINIT
BE1CHKPT  DL/I                             UP   NOTOPEN
BE1PARTA                                   UP   STOPPED, NOTOPEN, NOTINIT
BE1PARTB                                   UP   STOPPED, NOTOPEN, NOTINIT
BE1PARTC                                   UP   STOPPED, NOTOPEN, NOTINIT
BE1PARTS                                   UP   STOPPED, NOTOPEN, NOTINIT
BE2ORDER  DL/I                             UP   NOTOPEN
BE2ORDRX  DL/I                             UP   NOTOPEN
BE2PARTS  DL/I                             UP   NOTOPEN
BE2PCUST  DL/I                             UP   NOTOPEN
BE3ORDER  DL/I                             UP   NOTOPEN
BE3ORDRX  DL/I                             UP   NOTOPEN
                                                                More...

F1=Help  F3=Exit  F4=Top  F6=Bottom  F7=Bkwd  F8=Fwd  F9=Retrieve  F12=Cancel
```

Figure 21. CICS-DBCTL IMS responses

Using DFHDBFK - The CDBM GROUP command file

Before you can use the **GROUP** command, CDBM requires a file in which all your predefined commands can be stored. This file, DFHDBFK, is the CDBM GROUP command file. It is a VSAM KSDS.

The DFHDBFK file is not required until you first attempt to use the /GROUP command.

Sample JCL to enable you to define this file and optionally to place some of your group definitions within it is provided in the *CICS System Definition Guide*. The example given there also shows you how to batch load this file if you want to do so.

You can load many definitions in this way, or load none. You can use whatever means you prefer to define this file and to maintain its content. One method of maintaining the content once the file is defined is provided by CDBM which offers a Maintenance function for the DFHDBFK file.

To use the /GROUP command functions of the CDBM transaction, add the data set name of the group command file to the definition of the DFHDBFK file within the DFHDBCTL group. For information about how to do this, see the *CICS Resource Definition Guide*.

Record layout in the CDBM GROUP command file

Each record in the DFHDBFK file may be up to 1428 characters long, as follows:

field	length	content	description
1	12	Group	a 12-character field containing your chosen name for this group. The acceptable characters are A-Z 0-9 $ @ and #. Leading or embedded blanks are not allowed, but trailing blanks are acceptable.
2	10	IMS Command	a 10-character field containing any of the IMS command verbs that are valid for CDBM (see *CICS IMS Database Control Guide* for details). Leading or embedded blanks are not allowed, but trailing blanks are acceptable. **Note:** The validity of the IMS command verb is not checked by CDBM. Invalid values will be reported by IMS when the command is attempted.
3	1406	IMS Command parameters	Up to 1406 characters of parameters appropriate to the chosen IMS command verb. (This will often consist of lists of databases.) **Note:** Wildcard characters may not be used in the parameters stored in the CDBM Group command file. This is unlike the other functions of the CDBM transaction which permit the use of wildcard characters to describe multiple similarly named databases.

The MAINTENANCE panel for DFHDBFK

If you press the Maintenance key (PF2) on the main CDBM panel, you get the panel shown in Figure 22 on page 71.

```
CDBM               CICS/DBCTL COMMAND GROUP MAINTENANCE
_         ACTION      A add  B browse  D delete  R read  U update
____________ GROUP          __________ IMS COMMAND
>                                                                    <
>                                                                    <
>                                                                    <
>                                                                    <
>                                                                    <
>                                                                    <
>                                                                    <
>                                                                    <
>                                                                    <
>                                                                    <
>                                                                    <
>                                                                    <
>                                                                    <
>                                                                    <
>                                                                    <
>                                                                    <
>                                                                    <
>                                                                    <

F1=Help  F3=Exit  F12=Cancel
```

Figure 22. CICS-DBCTL Group Maintenance panel

Input fields

The input fields are:

- Action
- Group
- IMS Command
- IMS Command parameters (between the > < marks).

Group, IMS Command and IMS Command parameters are described in the section "Record layout in the CDBM GROUP command file" on page 70

The Action field will accept one of the following:

A Add

Add a new record to the DFHDBFK file. If the key already exists, the Add fails.

Note: To Add a record that is very similar to an existing record, but which has a different key, you may find it helpful to Read the existing record, modify the displayed fields, and then Add this new record.

B Browse

Displays the contents of the command file, record by record. Specify any key (or none) to indicate where you want the browse to start. Each time you press ENTER, Browse moves on to the next record. At the end of the file you will be prompted to wrap around to the start of the file. You can accept this or not as you prefer. Incomplete keys, and unknown keys are also acceptable as start points. If no key is provided, the browse starts at the first record in the file.

If you have used Browse to locate a specific record for deletion or for update, remember to use Read before either Delete or Update.

D Delete

Delete a record from the DFHDBFK file. A Delete must be immediately preceded by a Read to lock the required record.

R Read

Read displays a specific record. Unlike Browse it does not operate on partial, or absent keys, and does not present the next record when you press ENTER.

Read is required before those actions (Delete and Update) which change an existing record. It locks that record against the possibility of being changed by another operator. This action also serves to help you confirm that the correct record has been selected.

A lock is released by exiting the CDBM maintenance panel, or by your next CDBM Maintenance action (whether that is the Update or Delete you had contemplated, or something different entirely).

U Update

Update a record in the DFHDBFK file. An Update must be immediately preceded by a Read to lock the required record.

You cannot update the key fields (GROUP and IMS COMMAND).

Note: Use Add to create a new key.

Note: In the descriptions above, *Key* refers to the 22 characters at the beginning of each record in the DFHDBFK file (namely the GROUP and IMS COMMAND).

Using the CDBM HELP panels

If you press the help key (PF1) from the CICS-DBCTL Operator Transaction panel, you get the panel shown in Figure 23.

```
 CDBM               Help: CICS-DBCTL Operator Transaction


CDBM           Use the transaction to send an IMS command to a DBCTL system.

Command        Type the command recognition character / followed by an IMS
               command and press enter to display responses.

Responses      Use the PF keys to page IMS responses.

Wildcards      * or + can be used within one database name.

End of volume  For /DBDUMP or /DBRECOVER commands only
               Choose one.
                 1. Do not force end of volume
                 2. Force end of volume
CICS APPLID
DBCTL ID       These are shown for information.

Maintenance    Enter the group command maintenance screen.

Example        /DIS DB DEPT* displays the status of several databases.

F3=Exit  F12=Cancel
```

Figure 23. CICS-DBCTL operator transaction help panel

If you press the help key (PF1) from the CICS-DBCTL Maintenance panel, you get the panel shown in Figure 24.

```
 CDBM                Help: CICS-DBCTL Operator Transaction


Maintenance    Store commands for issuing from the CDBM screen.

GROUP          Enter the group you want to store a command in.

IMS COMMAND    Enter a valid IMS command to execute with the supplied data.

ACTION         A - Add a command to the command file.
               B - Browse the contents of the command file.
               D - Delete a command, only after it has been read.
               R - Read a command from the file.
               U - Update a command, only after it has been read.

               Issue commands from the main screen in the format
               /GROUP group command.

Example        /GROUP SAMPLE DIS shows information for the databases in SAMPLE.



F3=Exit  F12=Cancel
```

Figure 24. CICS-DBCTL Maintenance help panel

Chapter 8. CEBR - temporary storage browse

Use the CEBR transaction to browse temporary storage queues and to delete them. You can also use CEBR to transfer a transient data queue to temporary storage in order to look at its contents, and to recreate the transient data queue when you have finished. The CEBR commands that perform these transfers allow you to create and delete transient data queues as well.

Remember that:

- browsing of Temporary Storage queues retrieves the next record, following whichever record has most recently been retrieved by *ANY active task.*

  This is can lead to confusion if for example an EXEC CICS READQ NEXT and a CEBR transaction attempt to work with the same Temporary Storage queue at the same time.
- transient data queue reads are destructive. If you read a transient data queue that is being used on an active system by applications, this is likely to cause problems.

You start the CEBR transaction by entering the transaction identifier CEBR, followed by the name of the queue you want to browse. For example, to display the temporary storage queue named CEBRS209, you enter:

```
CEBR CEBRS209
```

If the queue name includes lower case characters, ensure that upper case translation is suppressed for the terminal you are using, and then enter the correct combination of upper and lower case characters.

CICS responds with a display of the queue, a sample of which is shown in Figure 25 on page 76:

© Copyright IBM Corp. 1982, 2011

```
CEBR  TS QUEUE CEBRS209 SYSID C142 RECORD     1 OF     3    COL     1 OF    22
ENTER COMMAND ===>
     **************************  TOP OF QUEUE  ************************
00001 000055001234000001S209
00002 000056003456000002S209
00003 000102000564000001S209
     ************************  BOTTOM OF QUEUE  **********************









PF1 : HELP              PF2 : SWITCH HEX/CHAR     PF3 : TERMINATE BROWSE
PF4 : VIEW TOP          PF5 : VIEW BOTTOM         PF6 : REPEAT LAST FIND
PF7 : SCROLL BACK HALF  PF8 : SCROLL FORWARD HALF PF9 : UNDEFINED
PF10: SCROLL BACK FULL  PF11: SCROLL FORWARD FULL PF12: UNDEFINED
```

Figure 25. Typical CEBR screen displaying temporary storage queue contents

Alternatively, you can start the CEBR transaction from CEDF. You do this by pressing PF5 from the initial CEDF screen (see Figure 42 on page 110) which takes you to the working-storage screen, and then you press PF2 from that screen to browse temporary storage (that is, you invoke CEBR). CEBR responds by displaying the temporary storage queue whose name consists of the four letters 'CEBR' followed by the four letters of your terminal identifier. (CICS uses this same default queue name if you invoke CEBR directly and do not supply a queue name.) The result of invoking CEBR without a queue name or from an EDF session at terminal S21A is shown in Figure 26 on page 77.

If you enter CEBR from CEDF, you will return to the EDF panel when you press PF3 from the CEBR screen.

```
CEBR  TS QUEUE CEBRS209 SYSID C142 RECORD     1 OF     0    COL     1 OF 10
ENTER COMMAND ===>                                                      2
*************************  TOP OF QUEUE  ******************************
************************  BOTTOM OF QUEUE  *****************************




                                                                        3









TEMPORARY STORAGE QUEUE CEBRS21A            IS EMPTY                    4
PF1 : HELP              PF2 : SWITCH HEX/CHAR     PF3 : TERMINATE BROWSE5
PF4 : VIEW TOP          PF5 : VIEW BOTTOM         PF6 : REPEAT LAST FIND
PF7 : SCROLL BACK HALF  PF8 : SCROLL FORWARD HALF PF9 : UNDEFINED
PF10: SCROLL BACK FULL  PF11: SCROLL FORWARD FULL PF12: UNDEFINED
```

Note: 1Header 2Command line 3Body 4Message line 5Menu of options

Figure 26. Typical CEBR display of default temporary storage queue

For information about the CEBR temporary storage browse transaction and guidance on using it, see the *CICS Application Programming Guide*.

Using the CEBR HELP panel

If you press the help key (PF1), the following panel is displayed:

```
These commands are available to you (abbreviations in UPPER CASE):
Find /string/               - Keyword optional. Final delimiter optional if
                              string has no blanks. Any other delimiter is OK.
Line line-number
Column column-number
Top
Bottom
TERMinal terminal-id        - Browse temp. storage queue for another terminal.
Queue temp-stg-queue        - Browse a named temp. storage queue
                              (name may be in hex - e.g., X'C134')
Sysid shared/remote sysid   - Browse shared or remote temp. storage queue
Put   transient-data-queue  - Copy current queue into a transient data queue.
Get   transient-data-queue  - Fetch a transient data queue for browsing.
PURGE                       - Destroy the current queue.
```

Chapter 9. CEBT - master terminal (alternate CICS)

Use CEBT to control an alternate CICS system in an extended recovery facility (XRF) environment.

For information about XRF, see the *CICS/ESA 3.3 XRF Guide*.

You can start this transaction only at the operating system console. Starting a transaction at the system console is described in “Using the system console” on page 10.

Minimum abbreviation of keywords

The CEBT command accepts as few characters of a keyword as needed to uniquely identify it within the request. An exception to this rule is the SHUTDOWN keyword, for which the minimum abbreviation is SHUT.

In the syntax displays on your screen and in this information, the minimum permitted abbreviation is given in uppercase characters, and the rest of the keyword in lowercase characters.

CEBT INQUIRE and SET commands

The **CEBT INQUIRE** and **CEBT SET** commands allow you to query or set:

- Auxiliary trace (“CEBT INQUIRE AUXTRACE” and “CEBT SET AUXTRACE” on page 83)
- The dump data set (“**CEBT INQUIRE DUMP**” on page 80 and “CEBT SET DUMP” on page 84)
- GTF tracing (“CEBT INQUIRE GTFTRACE” on page 81 and “CEBT SET GTFTRACE” on page 85)
- Internal tracing (“**CEBT INQUIRE INTTRACE**” on page 82 and “CEBT SET INTTRACE” on page 86)
- Surveillance or ‘tracking’ of the active CICS system by the alternate CICS system (“**CEBT INQUIRE SURVEILLANCE**” on page 82 and “CEBT SET SURVEILLANCE” on page 86)
- Takeover from the active CICS system by the alternate CICS system (“**CEBT INQUIRE TAKEOVER**” on page 83 and “CEBT SET TAKEOVER” on page 87).

CEBT INQUIRE AUXTRACE

Return information about the status of auxiliary trace.

Description

Auxiliary trace entries are made to a pair of CICS-controlled BSAM data sets, ‘A’ and ‘B’.

CEBT INQUIRE AUXTRACE tells you:

- Whether the auxiliary trace data set is open or closed
- Which of the auxiliary trace data sets is active, Cur(A) or Cur(B)
- Whether an auxiliary trace is in progress (started) or not (stopped).

© Copyright IBM Corp. 1982, 2011

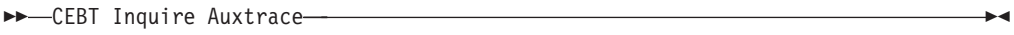
For background information about auxiliary tracing, see the *CICS Problem Determination Guide*.

Input

Press the Clear key and type `CEBT INQUIRE AUXTRACE` (the minimum abbreviation is `CEBT I AUX`).

To change various attributes, use the **CEBT SET AUXTRACE** command.

CEBT INQUIRE AUXTRACE



Displayed fields

All
Automatic switching occurs as necessary until the end of this CICS session without the need for operator intervention.

Auxiliary
identifies this panel as relating to auxiliary trace.

Current(A|B)
displays a 1-character identifier of the current auxiliary data set, which can be 'A' or 'B'. This value is blank if there is no auxiliary trace data set.

Next
Automatic switching occurs when the current trace data set is full, but not subsequently.

Noswitch
Switching does not occur without operator intervention.

Pause
Auxiliary tracing has stopped, but the trace data set has been left open. A subsequent START request causes trace entries to be written immediately following those that were written before the PAUSE request.

Start
Auxiliary tracing is in progress.

Stop
Auxiliary tracing has stopped.

Switch
displays the information that an immediate switch is being made from the current auxiliary trace data set to the alternate trace data set.

CEBT INQUIRE DUMP

Retrieve information about the CICS dump data sets.

Description

CEBT INQUIRE DUMP tells you which dump data set (A or B) is currently open.

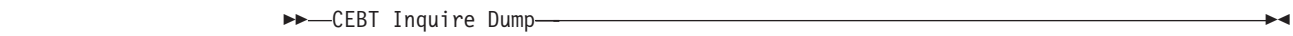
For general information about dumps, see the *CICS Problem Determination Guide*..

Input

Press the Clear key and type `CEBT INQUIRE DUMP` (the minimum abbreviation is `CEBT I DU`).

To change various attributes, use the **CEBT SET DUMP** command.

CEBT INQUIRE DUMP



Displayed fields

Autoswitch
The switch occurs **only** when the active dump data set is full, but not subsequently.

Closed
The active CICS dump data set is closed.

Currentdds(*value***)**
displays a 1-character indicator as to which of the two dump data sets is active. This can be A or B.

Dump
indicates that this panel relates to a dump data set inquiry.

Noautoswitch
When the active dump data set is full, there is not to be an automatic switch to the inactive dump data set.

Note: Because this is a 'negative' attribute, the field appears blank.

Open
The active CICS dump data set is open.

Switch
The currently active dump data set is to become inactive, and the currently inactive data set is to become active.

CEBT INQUIRE GTFTRACE

Show the GTF status.

Description

CEBT INQUIRE GTFTRACE tells you whether CICS tracing to the MVS Generalized Tracing Facility (GTF) is started or stopped.

This command is intended for use at terminals that do not support the CETR full-screen tracing transaction.

For information about GTF tracing, see the *CICS Problem Determination Guide* and the *CICS Performance Guide*.

Input

Press the Clear key and type `CEBT INQUIRE GTFTRACE` (the minimum abbreviation is `CEBT I G`).

To change various attributes, use the **CEBT SET GTFTRACE** command.

CEBT INQUIRE GTFTRACE

Displayed fields

Gtftrace
indicates that this panel relates to a GTFTRACE inquiry.

Start
CICS tracing to the MVS Generalized Tracing Facility (GTF) is in progress.

Stopped
CICS tracing to the MVS Generalized Tracing Facility (GTF) is not in progress.

CEBT INQUIRE INTTRACE

Show the status of CICS internal tracing.

Description

CEBT INQUIRE INTTRACE tells you whether CICS internal tracing is in progress.

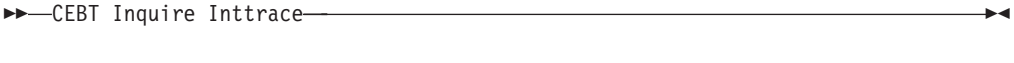
This command is intended for use at terminals that do not support the full-screen tracing transaction CETR.

Input

Press the Clear key and type `CEBT INQUIRE INTTRACE` (the minimum abbreviation is `CEBT I IN`).

To change various attributes, use the **CEBT SET INTTRACE** command.

CEBT INQUIRE INTTRACE



Displayed fields

Inttrace
indicates that this panel relates to an INTTRACE inquiry.

Start
CICS internal tracing is in progress.

Stopped
CICS internal tracing is not in progress.

CEBT INQUIRE SURVEILLANCE

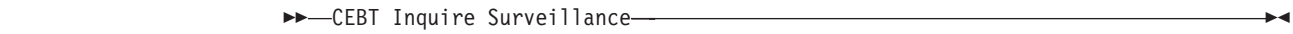
Display whether surveillance (or tracking) of the active CICS system by the alternate CICS system is ON or OFF.

Input

Press the Clear key and type `CEBT INQUIRE SURVEILLANCE` (the minimum abbreviation is `CEBT I SU`).

To change various attributes, use the **CEBT SET SURVEILLANCE** command.

CEBT INQUIRE SURVEILLANCE



For more information about surveillance, see the *CICS/ESA 3.3 XRF Guide*.

CEBT INQUIRE TAKEOVER

Display whether any takeover from the active CICS system by the alternate CICS system is to be automatic, manual, or command-driven.

Input

Press the Clear key and type `CEBT INQUIRE TAKEOVER` (the minimum abbreviation is `CEBT I T`).

To change various attributes, use the **CEBT SET TAKEOVER** command.

CEBT SET AUXTRACE

Function

Change auxiliary tracing options.

Description

You can start, stop, or pause tracing, and specify that automatic switching of the auxiliary data set is to occur when needed. You can use the switch option to change the data set.

Instead of using the **SET** command, you can overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 128).

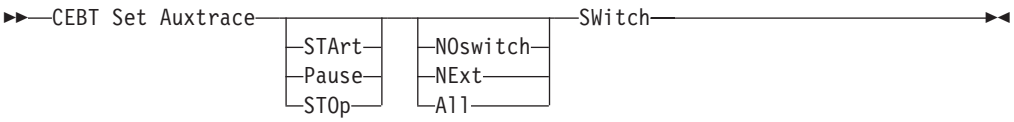
For information about using traces, see the *CICS Problem Determination Guide*.

Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEBT SET AUXTRACE` (the minimum abbreviation is `CEBT S AUX`). You get a display that lists the current status, similar to that obtained by **CEBT INQUIRE AUXTRACE**. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEBT SET AUXTRACE` (`CEBT S AUX`) followed by one or more attribute settings that you wish to change. For example, `CEBT S AUX STO` will stop auxiliary tracing if it is already running.

CEBT SET AUXTRACE



Options

All
Specifies that automatic switching between the two auxiliary trace data sets is to occur as necessary until the end of this CICS session, without the need for operator intervention.

NExt
Automatic switching of the auxiliary trace data set is to occur when the current trace data set is full, but not subsequently.

NOswitch
Switching between the auxiliary trace data sets is not to occur without operator intervention.

Pause
Auxiliary tracing is to stop, but the data set is to remain open. A subsequent START request causes trace entries to be written immediately following those that were written before the PAUSE request.

STArt
CICS is to start auxiliary tracing and open the auxiliary trace data set if it is currently closed.

STOp
CICS is to stop auxiliary tracing and close the auxiliary trace data set. A subsequent START request causes new trace entries to be written at the start of the data set, thereby overwriting the trace entries that were written before the STOP request.

SWitch
specifies that an immediate switch from the current auxiliary trace data set is to be made to the alternate trace data set.

CEBT SET DUMP

Change the attributes of a CICS dump data set.

Description

Using **CEBT SET DUMP**, you can open or close the active CICS dump data set, and specify whether or not the automatic switch to the inactive dump data set is to occur when the active dump data set is full. Instead of using the SET command, you can overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 128).

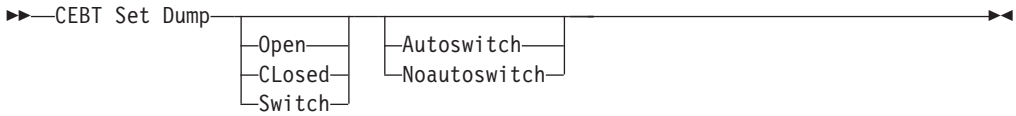
For information about using dumps, see the *CICS Problem Determination Guide*.

Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type CEBT SET DUMP (the minimum abbreviation is CEBT S DU). You get a display that lists the current status, similar to that obtained by **CEBT INQUIRE DUMP**. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type CEBT SET DUMP (CEBT S DU) followed by one or more attribute settings that you wish to change. For example, CEBT S DU A will cause an automatic switch to the inactive dump data set if 'noautoswitch' was previously specified.

CEBT SET DUMP



Options

Autoswitch
The switch occurs **only** when the active dump data set is full, but not subsequently.

CLosed
The active CICS dump data set is closed.

Noautoswitch
When the active dump data set is full, there is not to be an automatic switch to the inactive dump data set.

Open
The active CICS dump data set is opened.

Switch
The currently active dump data set is to become inactive, and the currently inactive data set is to become active.

CEBT SET GTFTRACE

Change the status of GTF tracing

Description

CEBT SET GTFTRACE allows you to switch CICS tracing to the MVS Generalized Tracing Facility (GTF) on and off. This command is intended for use at terminals that do not support the CETR full-screen tracing transaction.

For information about using GTF tracing, see the *CICS Problem Determination Guide*.

Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type CEBT SET GTFTRACE (the minimum abbreviation is CEBT S G). You get a display that lists the current status, similar to that obtained by **CEBT INQUIRE GTFTRACE**. You can then tab to the highlighted or blank fields and overtype them with the required values.

- Type `CEBT SET GTFTRACE` (`CEBT S G`) followed by one or more attribute settings that you wish to change. For example, `CEBT S G STO` will stop GTF tracing if it is already running.

CEBT SET GTFTRACE

Options

STArt
CICS tracing to the MVS Generalized Tracing Facility (GTF) is started.

STOp
CICS tracing to the MVS Generalized Tracing Facility (GTF) is stopped.

CEBT SET INTTRACE

Switch CICS internal tracing on or off.

Description

This command is intended for use at terminals that do not support the full-screen tracing transaction CETR.

Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEBT SET INTTRACE` (the minimum abbreviation is `CEBT S IN`). You get a display that lists the current status, similar to that obtained by **CEBT INQUIRE INTTRACE**. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEBT SET INTTRACE` (`CEBT S IN`) followed by one or more attribute settings that you wish to change. For example, `CEBT S IN STO` will stop internal tracing if it is already running.

CEBT SET INTTRACE

Options

STArt
CICS internal tracing is started.

STOp
CICS internal tracing is stopped.

CEBT SET SURVEILLANCE

Set surveillance (or tracking) of the active CICS by the alternate CICS on or off.

Syntax

CEBT SET SURVEILLANCE

Options

OFf
Surveillance of the active CICS system by the alternate CICS is to be stopped.

ON
Surveillance of the active CICS system by the alternate CICS is to be started.

For more information about surveillance, see the *CICS/ESA 3.3 XRF Guide*.

CEBT SET TAKEOVER

Set takeover from the active CICS system by the alternate CICS system to be automatic, manual, or command-driven.

Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEBT SET TAKEOVER` (the minimum abbreviation is `CEBT S T`). You get a display that lists the current status, similar to that obtained by **`CEBT INQUIRE TAKEOVER`**. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEBT SET TAKEOVER` (`CEBT S T`) followed by one or more attribute settings that you wish to change. For example, `CEBT S T M` changes takeover to a manual operation if it was previously specified to be automatic.

CEBT SET TAKEOVER

Options

Automatic
Takeover is to occur automatically.

Command
Takeover is to be by command.

Manual
Takeover is to be manual.

The *CICS/ESA 3.3 XRF Guide* tells you more about these command options.

CEBT PERFORM commands

The **CEBT PERFORM** commands allow you to:

- Shut down the alternate CICS system ("**CEBT PERFORM SHUTDOWN**")
- Request a dump of the alternate CICS and continue processing ("**CEBT PERFORM SNAP**")
- Cause the alternate CICS to take over from the active CICS ("CEBT PERFORM TAKEOVER").

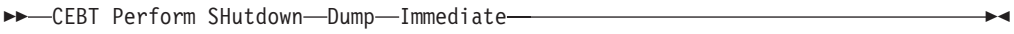
CEBT PERFORM SHUTDOWN

Shut down the alternate CICS system in an XRF complex.

Description

If you enter **CEBT PERFORM SHUTDOWN**, all active system tasks are allowed to finish, and the alternate CICS system signs off normally from the CICS availability manager (CAVM) data set.

CEBT PERFORM SHUTDOWN



Options

Dump
causes a dynamic storage dump to be produced when the shutdown is complete.

Immediate
causes the alternate CICS system to be shut down immediately, terminating all active tasks. The alternate CICS signs off abnormally from the CAVM data set. This can cause the primary (active) CICS system to abend if tasks are still running.

CEBT PERFORM SNAP

This command causes a "snapshot" dump of the alternate CICS system to be taken whilst CICS continues to run. All **CEBT PERFORM SNAP** requests result in an SDUMP on an MVS SYS1.DUMP*nn* data set. The use of **CEBT PERFORM SNAP** prevents all other tasks from running on the alternate system, and could delay takeover or recovery after takeover.

Syntax

CEBT PERFORM SNAP

CEBT PERFORM TAKEOVER

The **CEBT PERFORM TAKEOVER** command initiates a takeover by the alternate CICS system, causing the active CICS system to sign off abnormally.

Syntax

CEBT PERFORM TAKEOVER

Options

Dump
sends a message to the active CICS system requesting it to take a system dump from the CAVM data set. This request is not honored if the active CICS fails before the dump can be taken.

Dump
sends a message to the active CICS system requesting it to take a system dump from the CAVM data set. This request is not honored if the active CICS fails before the dump can be taken.

Chapter 10. CECI - command-level interpreter

Use CECI, which invokes the CICS command-level interpreter, to enter an **EXEC CICS** command, check its syntax, and modify it if necessary. In most cases, you can also process the command.

For programming information on each **EXEC CICS** command, see the *CICS Application Programming Reference* or the *CICS System Programming Reference*.

For details of how to use the CECI transaction, see in the *CICS Application Programming Guide*, which includes an example of a complete session with CECI.

Start the transaction by entering CECI on the command line of the screen, as follows:

```
CECI
```

Press ENTER. You get a list of all **EXEC CICS** commands.

```
 STATUS:  ENTER ONE OF THE FOLLOWING

ABend        DISAble      ISsue        RELease      SUspend
ACquire      DISCard      Journal      RESEtbr      SYncpoint
ADdress      DUmp         LInk         RESYnc       Trace
ALlocate     ENAble       LOad         RETRieve     Unlock
ASKtime      ENDbr        Monitor      RETUrn       Verify
ASSign       ENQ          PErform      REWrite      WAIT
BIf          ENTer        POInt        ROute        WAITCics
BUild        EXtract      POP          SENd         WRITE
CAncel       FEpi         POSt         SET          WRITEQ
CHange       FOrmattime   PURge        SIGNOFf      Xctl
COLlect      FREE         PUSh         SIGNON
CONNect      FREEMain     Query        SPOOLClose
CONVerse     GDs          READ         SPOOLOpen
DELAy        GEtmain      READNext     SPOOLRead
DELETE       Handle       READPrev     SPOOLWrite
DELETEQ      IGnore       READQ        START
DEQ          INquire      RECeive      STARTBr



PF 1 HELP 2 HEX 3 END 4 EIB 5 VAR 6 USER            9 MSG
```

Figure 27. CECI transaction: initial screen

Note: If you then enter the command or its abbreviation on the top line and press ENTER, you get a complete display of the syntax for that command. For example, Figure 28 on page 92 shows the screen displayed for the **EXEC CICS RETRIEVE** command.

© Copyright IBM Corp. 1982, 2011

```
RETR
STATUS:  ABOUT TO EXECUTE COMMAND                          NAME=
 EXEC CICS  RETRieve
  < Set() | Into() >
  < Length() >
  < RTRansid() >
  < RTErmid() >
  < Queue() >
  < Wait >









PF 1 HELP 2 HEX 3 END 4 EIB 5 VAR 6 USER 7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 28. CECI transaction: RETRieve command screen

Alternatively, you can enter CECI together with the name of the command in which you are interested, and bypass the full list of commands; for example:

```
CECI asktime
```

In both cases, you can generally go on to check the syntax of the command, change it if necessary, and then process the command.

Many CICS system programming commands use CVDA values, and CECI shows numeric equivalents for these. See the *CICS System Programming Reference* for programming information, including a “translation” of these numeric values to their alphabetic values.

Using the CECI HELP panel

If you press the help key (PF1), the following panel is displayed:

```
 GENERAL HELP INFORMATION

 Enter command on the first line and press ENTER (after returning from HELP).
 Options can be abbreviated to the minimum to make them unique.
 ? before command gives COMMAND SYNTAX CHECK and prevents execution.
 To expand a value or variable to full screen,
   position cursor using TAB key and press ENTER.

 PF KEY HELP INFORMATION

   PF01   HELP  HELP INFORMATION
   PF02   HEX   SWITCH HEX/CHAR
   PF03   END   END SESSION
   PF04   EIB   EXEC INTERFACE BLOCK
   PF05   VAR   VARIABLES
   PF06   USER  USER DISPLAY
   PF07   SBH   SCROLL BACK HALF
   PF08   SFH   SCROLL FORWARD HALF
   PF09   MSG   MESSAGES
   PF10   SB    SCROLL BACK
   PF11   SF    SCROLL FORWARD
   PF12         UNDEFINED
USE ENTER TO RETURN
```

Figure 29. CECI transaction: general HELP panel

Command-level syntax-checking transaction (CECS)

Use the CECS transaction, which also invokes the command-level interpreter, to check the syntax of an **EXEC CICS** command, but not to process it.

Start the transaction by entering CECS on the command line of the screen, as follows:

```
CECS
```

Press ENTER. You get a list of all the **EXEC CICS** commands. This is similar to the CECI screen shown in Figure 27 on page 91. Enter the command or its abbreviation at the top of the screen (next to the ?) and press ENTER and you get a complete display of the syntax for that command.

Alternatively, you can enter CECS together with the name of the command in which you are interested, and bypass the full list of commands:

```
CECS command
```

For information how to use the CECI and CECS transactions, see the *CICS Application Programming Guide*.

Chapter 11. CEDA - resource definition online

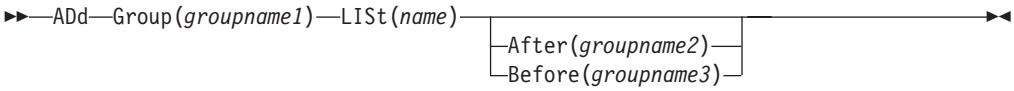
Use CEDA to define resources to your CICS system while the system is running. With CEDA, you can update both the CICS system definition (CSD) file and the running CICS system.

This section briefly describes CEDA and it's syntax. CEDA is described in detail in the *CICS Resource Definition Guide*.

Command syntax

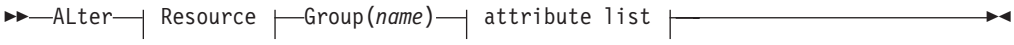
CEDA

ADD



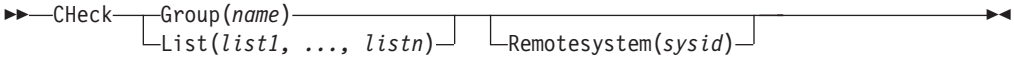
Note: Consult the *CICS Resource Definition Guide* for information about the attributes applicable to each resource type.

ALTER

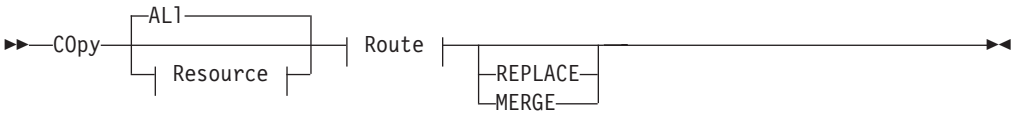


APPEND

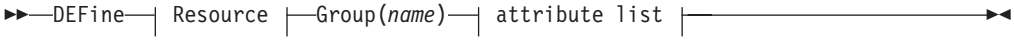
CHECK



COPY



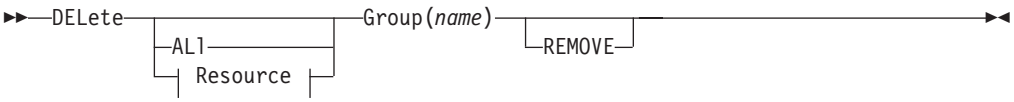
DEFINE



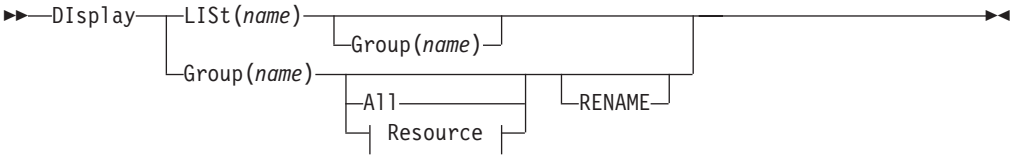
Note: Consult the *CICS Resource Definition Guide* for information about the attributes applicable to each resource type.

© Copyright IBM Corp. 1982, 2011

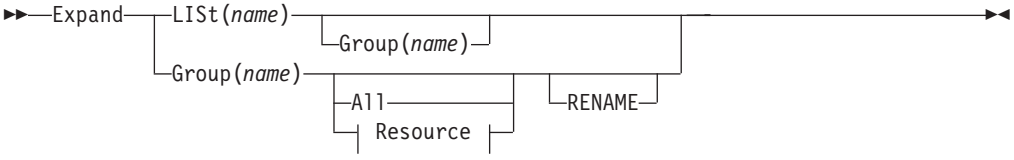
DELETE



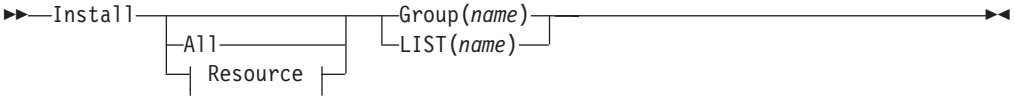
DISPLAY



EXPAND

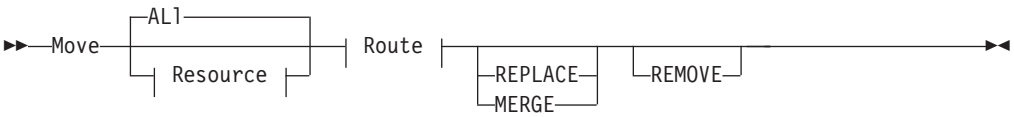


INSTALL

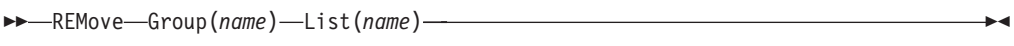


LOCK

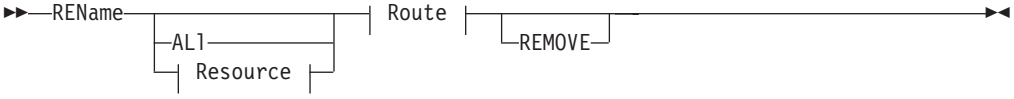
MOVE



REMOVE

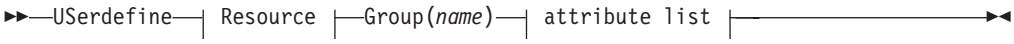


RENAME



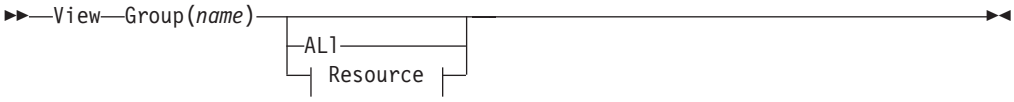
UNLOCK

USERDEFINE



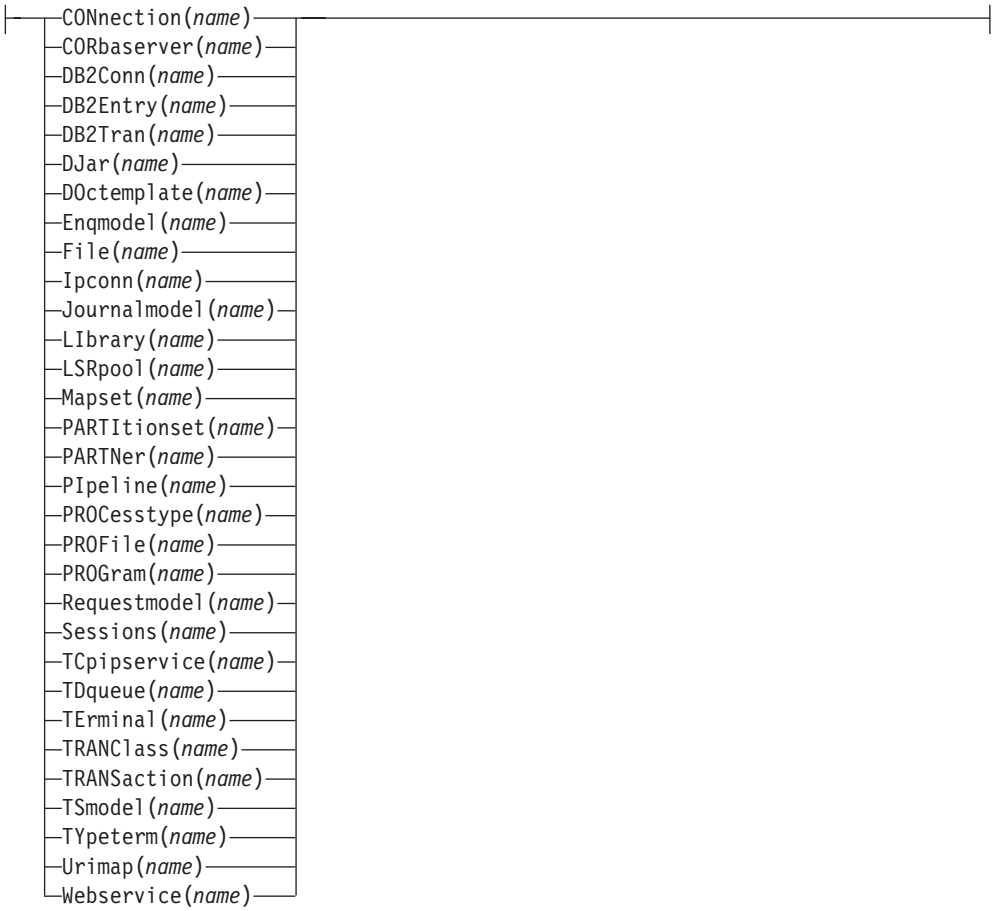
Note: Consult the *CICS Resource Definition Guide* for information about the attributes applicable to each resource type.

VIEW

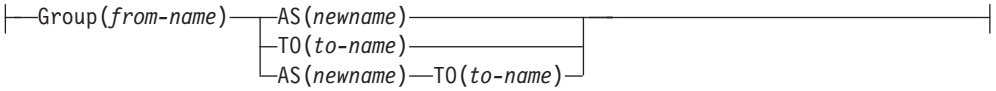


CEDA continued

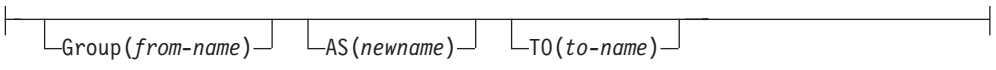
Resource:



Route (Copy and Move):



Route (Rename):



attribute list:

You can go directly to the working screen to make your changes by typing:

```
CEDA function resource_type
```

For example:

```
CEDA DEF CONN
```

The CEDA keywords are as follows:

AFTER
Controls the placing of a group name in a list.

ALL
All resources are to be copied, deleted, displayed, installed, moved, or renamed.

AS
The new name specified for a definition.

BEFORE
Controls the placing of a group name in a list.

CONNECTION
The remote system that your CICS system communicates with, using ISC or MRO.

CORBASERVER
Defines the execution environment for enterprise beans and stateless CORBA objects.

DB2CONN
Defines the CICS DB2® connection.

DB2ENTRY
Defines resources to be used by a transaction or group of transactions when accessing DB2.

DB2TRAN
Associates a transaction ID with a DB2ENTRY.

DJAR
Defines a deployed JAR file.

DOCTEMPLATE
Defines a document templates to CICS. Document templates allow you to perform variable substitution on documents in a manner similar to what is done by BMS for 3270 screens.

ENQMODEL
Use of ENQMODEL definitions provides an enqueue mechanism that serializes access to a named resource across a specified set of CICS regions operating with a sysplex.

FILE
The FILE definition describes to CICS file control the physical and operational characteristics of a file.

GROUP
Every resource definition must have a GROUP name. The resource definition becomes a member of the group and is installed in the CICS system when the group is installed. The GROUP name can be up to eight characters in length.

JOURNALMODEL
The name of a journal model containing specifications of journal names. An MVS™ log stream name is associated with each journal model entry.

LIST
The name of the list for which a specified group is to be added, appended, checked, displayed, expanded, installed, locked, removed, or unlocked.

LSRPOOL
The local shared resources (LSR) pool is a reserve of data buffers, strings, and hiperspace buffers that VSAM uses when processing access requests for certain files.

MAPSET
A series of different BMS maps.

PARTITIONSET
A table that describes to CICS how a display screen is to be partitioned.

PARTNER
The PARTNER definition enables CICS application programs to communicate via APPC protocols to a partner application program running on a remote logical unit. PARTNER is one of the four resources (the other three are PROFILE, CONNECTION, and SESSION) that you must specify to allow the System Application Architecture (SAA) communications interface to be used.

PIPELINE
A PIPELINE resource definition is used when a CICS application is in the role of a Web service provider or requester. It provides information about the message handler programs that act on a service request and on the response.

PROCESSTYPE
Use the CICS business transaction services (BTS) API to define and execute complex business applications called *processes*.

PROFILE
A group of programs that control a number of transactions or partners.

PROGRAM
The name of the program that is to control a given transaction.

REQUESTMODEL
A REQUESTMODEL resource definition provides the connection between an Internet Inter-ORB Protocol (IIOP).

SESSIONS
The logical link of two systems that communicate using ISC or MRO.

TCPIPSERVICE
Use this resource definition to define which TCP/IP services are to use CICS internal sockets support. The internal CICS services that can be defined are IIOP and the CICS Web Interface.

TDQUEUE
The definition of each transient data queue used in your CICS system. Queue types can be intrapartition, extrapartition, or indirect. Remote queues cannot be specified explicitly; see the the *CICS Resource Definition Guide* for more information about defining transient data queues to be remote.

TERMINAL
The definition of each terminal used in your CICS system. A terminal can be any device linked to your system, for example, visual display units, printers, facsimile machines. To simplify matters, it is possible to group resources that have common features by using TYPETERM and through AUTOINSTALL.

TO
The new group to which a definition is to be copied.

TRANCLASS
The name of a user-defined class of transactions.

TRANSACTION
The name of the transaction. Within the transaction definition, you can specify options such as transaction priority and length of the transaction work area. You also add to the transaction name the names of related resources: PROGRAM, PROFILE, and PARTITIONSET.

TSMODEL
The TSMODEL resource allows you to specify a TS queue name prefix, and associate attributes with that name. You can also map names directly to a shared TS pool (without the need for a shared SYSID).

TYPETERM
Some of the common attributes of your terminals. Which has to be defined as part of the terminal definition.

URIMAP
A URIMAP is a resource definition that matches the URIs of HTTP or Web service requests, and provides information of how to process the requests.

WEBSERVICE
A WEBSERVICE resource defines aspects of the run time environment for a CICS application program deployed in a Web services setting, where the mapping between application data structure and SOAP messages has been generated using the CICS Web services assistant.

CSD compatibility panel

Figure 30 shows a normal ALTER panel for the bottom part of the REQUESTMODEL definition, with the OMGINTERFACE, OMGMODULE, and OMGOPERATION attributes, that are now obsolete, protected as indicated by the colon symbol (:).

```
  ALTER                                                    RELEASE 0650
  OVERTYPE TO MODIFY
.
.
.
  TRANSACTION ATTRIBUTES
   TRansid      ==> CIRP

  PREVIOUS ATTRIBUTES
   OMGModule    :
   OMGInterface :
   OMGOperation :
```

Figure 30. The ALTER panel for REQUESTMODEL: normal mode

You should note that a large field, such as Module, when it is not fully displayed on the current screen, is also protected:

```
 Requestmodel   : ........
 Group          : ........
 Description  ==> ....................................
 Corbaserver  ==> *
 TYpe         ==> Generic            Corba | Ejb | Generic
EJB PARAMETERS
 Beanname     ==> *
              ==>
              ==>
              ==>
 INTFacetype  ==> Both               Both | Home | Remote
CORBA PARAMETERS
 Module       :
              :
```

Figure 31. The DEFINE panel for REQUESTMODEL

However, when you scroll forward, the whole field is displayed and becomes enabled:

```
CORBA PARAMETERS
 Module       ==> *
              ==>
              ==>
              ==>
 INTErface    ==> *
              ==>
              ==>
              ==>
COMMON PARAMETERS
 OPeration    ==> *
              ==>
              ==>
              ==>
TRANSACTION ATTRIBUTES
 TRansid      ==> CIRP

PREVIOUS ATTRIBUTES
 OMGModule    :
 OMGInterface :
 OMGOperation :
```

Figure 32. The DEFINE panel for REQUESTMODEL

See the *CICS Resource Definition Guide* for details of using CEDA. See the *CICS Resource Definition Guide* also for background information about the CSD compatibility panel.

Use the PF2 key to change to the compatibility mode panel shown in Figure 33 on page 103. The OMGINTERFACE, OMGMODULE, and OMGOPERATION attributes are now shown with the ==> symbols, indicating that they can be modified. Use the PF2 key again to return to the normal panel.

```
ALTER                                      COMPATIBILITY MODE
.
.
.
TRANSACTION ATTRIBUTES
 TRansid      ==> CIRP

PREVIOUS ATTRIBUTES
 OMGModule    ==>
 OMGInterface ==>
 OMGOperation ==>
```

Figure 33. The ALTER panel for REQUESTMODEL: compatibiity mode

Note that it is not possible to have both current and obsolete attributes in the same request model definition. If you enter OMG attribute values, you must remove all attribute values from previous versions. For more information, see the *CICS Resource Definition Guide*.

Security checking

The scope of a particular CEDA command can be limited by a combination of several levels of security checking.

The normal transaction-attach security is described in the *CICS RACF Security Guide*. CEDA calls the exec interface programs to process the user's commands, in much the same way as CECI or a user's command-level program would. The resource security and the command security used by the command-level programming interface are thus applicable to the master terminal user.

For programming information about command security checking, see the *CICS System Programming Reference*.

CEDA HELP panels

If you press the help key (PF1), the following panels are displayed:

```
                        CEDA Help Selection

This panel allows you to see a brief description of the CEDA transaction.
For a complete description see Resource Definition (Online)  (SC33-0666).

Select one of the following topics:

                1 Command Summary
                2 Resources, Groups and Lists
                3 Using the commands
                4 Expand and Display
                5 Messages
                6 Defaults and Userdefine
                7 PF keys

  Selection ==>









 Press Enter or any PF key to return
```

Figure 34. CEDA transaction: initial HELP panel

```
                           Command Summary

Resource management commands:
  DEFINE        creates a resource definition (see Topic 6 for USERDEFINE)
  ALTER,VIEW    modify and display the attributes of a resource (or resources)
  COPY          creates new resources from existing definitions
  DELETE        destroys resource definitions
  MOVE,RENAME   change the names and/or groupings of resources

List management commands:
  ADD           creates or extends a list
  REMOVE        reduces or destroys a list
  APPEND        copies a list or combines lists

General purpose commands:
  CHECK         cross-checks definitions in a group or list
  DISPLAY       shows the names of groups or lists on the CSD file
  EXPAND        shows the contents of groups or lists
  INSTALL       dynamically adds resources to the running CICS system
  LOCK,UNLOCK   control access to a group or list

CEDB does not have INSTALL. CEDC has DISPLAY, EXPAND and VIEW only.

 Press Enter or any PF key to return to Help Selection Panel
```

Figure 35. CEDA transaction: HELP Panel 1

```
                    Resources, Groups and Lists

You use CEDA to create and modify resource definitions. Using the DEFINE
command, you specify a resource's type, name and attributes, which are
then stored on the CICS System Definition (CSD) file.

You can see what types of resource there are by using DEFINE on its own
as a command. Similarly you can see what attributes any resource may have
by adding just the type of resource, as in, for example, DEFINE PROGRAM.

Each resource must belong to a GROUP, which is a collection of resources,
usually related in some way. A group of resources can be installed on
your running CICS system.

A LIST is a collection of group names, and can be used to specify large
numbers of resources during a cold start.

Note that program P, say, may be defined in more than one group. Such
definitions are separate resources and may have different attributes.
By contrast the same group names in different lists refer to the same group.
The DELETE command destroys a resource, but REMOVE does not destroy a group.
A group has no attributes and need not even exist to be used in a list.

 Press Enter or any PF key to return to Help Selection Panel
```

Figure 36. CEDA transaction: HELP Panel 2

```
                        Using the commands

You type CEDA commands on the first line of the screen and press ENTER.
You will then see a panel that shows your command in detail, and the
results of its execution. You can then either modify the panel to
execute a similar command with new values or type a different command
on the top line.

You can see the syntax of a command, without executing it, by typing ?
in front of the command.

You can shorten command keywords as much as you like provided the result
remains unique. Thus ALT and AL both mean ALTER but A is invalid
because of ADD. The minimum number of letters you can use is shown
in upper case.

You can specify generic names in some commands, by using * and +.
* means any number of characters, + means any single character.
Thus PROGRAM(P*) refers to all programs whose names begin with P.

Current values for GROUP and LIST are kept and are used when either
keyword is omitted from commands other than DISPLAY and EXPAND LIST.

 Press Enter or any PF key to return to Help Selection Panel
```

Figure 37. CEDA transaction: HELP Panel 3

```
                        Expand and Display

The commands you can use on each kind of panel are as follows:

  EXPAND GROUP panel  - Alter, Copy, Delete, Install, Move, Rename, View
  EXPAND LIST panel   - Add, Remove
  DISPLAY GROUP panel - Check, Expand, Install , Lock, Unlock
  DISPLAY LIST panel  - Append, Check, Expand, Lock, Unlock

All these commands operate on the thing beside which you enter them.
ALTER means ALTER PROGRAM(P) GROUP(G), if these are the object and
group values on the line. In this case, since no attributes are changed,
you will see a display of the object which you can then overtype.
The EXPAND command on a DISPLAY panel also results in a new panel.

You can enter as many commands as you like at one time and you can
use = to mean the same command as the last one.

The RENAME option of EXPAND GROUP gives a panel on which you can change the
names of objects directly, by overtyping the name fields displayed.
When you change a name field a RENAME command is put in the corresponding
command field, and causes anything entered there to be ignored.

 Press Enter or any PF key to return to Help Selection Panel
```

Figure 38. CEDA transaction: HELP Panel 4

```
                             Messages

Single messages appear near the bottom of panels. If there is more than one
message a summary appears instead. PF9 shows the details of such a summary.

Messages are preceded by a single letter indicating the severity:
   I-Informatory   W-Warning     E-Error    S-Severe

Commands with only I or W-level messages are executed. E-level messages
are given for errors that CEDA attempts to correct. Such commands can
be executed by pressing ENTER without making any changes. S-level
messages require you to correct the command.


For a command executed on an EXPAND or DISPLAY panel you can see the
messages by using ? in the command field or by means of the cursor and PF9.
Commands with errors that have apparently been fixed (E-level messages)
must still be corrected by you.
You can correct a command on the message panel or on the original panel.




 Press Enter or any PF key to return to Help Selection Panel
```

Figure 39. CEDA transaction: HELP Panel 5

```
                      Defaults and Userdefine

When you DEFINE a resource any attributes you do not specify are
defaulted. You cannot change the defaults that DEFINE uses.
You can however DEFINE "default resources" with whatever values you
like and then create new resources using USERDEFINE.

You DEFINE resources called USER in a group called USERDEF, giving
each one the values you would like to have as defaults. These are
now your default resources.

USERDEFINE will then behave just like DEFINE except that it will get
default values from the appropriate default resource. If a default
resource does not exist then USERDEFINE fails.

This facility is restricted to initial values only. Default values
are also given by CEDA to attributes you remove from a resource, by
overtyping with blanks for instance. These defaults are the same as
those used for the DEFINE command and you cannot change them.




 Press Enter or any PF key to return to Help Selection Panel
```

Figure 40. CEDA transaction: HELP Panel 6

```
                              PF keys

The PF keys do the following:

     1 HELP    Gives the initial help panel
     2 COM     Selects and deselects compatibility mode
     3 END     Terminates the CEDA transaction if no data has been entered
     4         Does nothing
     5         Does nothing
     6 CRSR    Moves the cursor to the command line or first input field
     7 SBH     Scrolls back half a screen
     8 SFH     Scrolls forward half a screen
     9 MSG     Displays the current set of messages
    10 SB      Scrolls back a full screen
    11 SF      Scrolls forward a full screen
    12 CNCL    Cancels changes not yet executed and returns to previous panel

When you press ENTER without having entered any data you will normally
be returned to the previous panel.

Positioning the cursor at a PF key and pressing ENTER will have the same
effect as pressing the key. PF13 to PF24 have the same effects as PF1 to PF12.

 Press Enter or any PF key to return to Help Selection Panel
```

Figure 41. CEDA transaction: HELP panel 7

CEDB and CEDC

Two further resource definition transactions, CEDB and CEDC, allow you to use some, but not all, of the functions of CEDA. These transactions are described in the CEDA transaction tutorial in the *Resource Definition Guide*.

CEDB

When you use the CEDB transaction, the INSTALL command is not available to you. This means that you can update the CSD, but not the running CICS system.

CEDC

The CEDC transaction allows you only to look at data on the CICS system definition (CSD) file. You cannot update either the CSD file or the running CICS system. The only options that are available are DISPLAY, EXPAND, and VIEW.

Chapter 12. CEDF and CEDX - the execution diagnostic facility

The CICS execution diagnostic facility (EDF) provides two transactions that you can use for testing application programs.

These transactions - CEDF and CEDX - enable you to test application programs interactively without having to supply special program testing procedures.

CEDF
Use CEDF to invoke EDF for testing application programs that are associated with user transactions initiated from a terminal.

CEDX
Use CEDX to invoke EDF for testing application programs that are associated with non-terminal transactions.

Using the CEDF transaction

CEDF allows you to intercept your application programs at the program's initiation, at each CICS command, and at the program termination. CEDF helps you to isolate and focus on problems in your application programs.

For information on how to use the CEDF transaction, see the *CICS Application Programming Guide*.

You must ensure that the EDF resource definitions are installed. These resource definitions are provided in the IBM-supplied group, DFHEDF. For information about installing the resource definitions, see the *CICS Resource Definition Guide*.

You must also ensure that the terminal from which you run EDF is in transceive (ATI/TTI) status.

Command syntax

CEDF

Command options

OFF
specifies whether the EDF screen is to be switched OFF. If you specify OFF you **must** enter the preceding comma, as shown in the following examples:

```
CEDF T33A
CEDF T33A ,OFF
CEDF ,OFF
```

© Copyright IBM Corp. 1982, 2011

ON
specifies whether the EDF screen is to be switched ON. The default is ON. If you specify ON you **must** enter the preceding comma, as shown in the following examples:

```
CEDF T33A
CEDF ,ON
```

sessionid
specifies the session identifier used to test a transaction attached across an MRO, APPC, or LU6.1 session. All CICS commands processed by the attached transaction are monitored.

sysid
specifies the system identifier used to test a transaction across an MRO or APPC session.

termid
specifies the identifier of the terminal on which the transaction to be tested is being run. If you do not provide an identifier, the terminal at which you enter the CEDF transaction is assumed.

Beginning an EDF session (single-screen mode)

To start an EDF session with a single terminal, clear the screen and enter the transaction identifier, CEDF. The EDF inputs and outputs are interleaved with those of the transaction. This is known as *single-screen mode*.

```
CEDF
```

When you press ENTER, the response is:

```
THIS TERMINAL: EDF MODE ON
```

Now clear the screen and enter the name of your transaction. Figure 42 shows the initial screen for a typical transaction using EDF.

```
 TRANSACTION: ACCT  PROGRAM: ACCT00  TASK: 0000054  APPLID: 00001349  DISPLAY: 00
 STATUS:  PROGRAM INITIATION

    EIBTIME      = 110343
    EIBDATE      = 89174
    EIBTRNID     = 'ACCT'
    EIBTASKN     = 54
    EIBTRMID     = '025L'

    EIBCPOSN     = 4
    EIBCALEN     = 0
    EIBAID       = X'7D'                                    AT X'00543F1E'
    EIBFN        = X'0000'                                  AT X'00543F1F'
    EIBRCODE     = X'000000000000'                          AT X'00543F21'
    EIBDS        = '........'
+   EIBREQID     = '........'

ENTER:  CONTINUE
PF1 : UNDEFINED           PF2 : SWITCH HEX/CHAR     PF3 : END EDF SESSION
PF4 : SUPPRESS DISPLAYS   PF5 : WORKING STORAGE     PF6 : USER DISPLAY
PF7 : SCROLL BACK         PF8 : SCROLL FORWARD      PF9 : STOP CONDITIONS
PF10: PREVIOUS DISPLAY    PF11: UNDEFINED           PF12: UNDEFINED
```

Figure 42. CEDF transaction: initial screen

Beginning an EDF session (dual-screen mode)

You can also use EDF in what is known as *dual-screen mode*; that is, using one terminal to monitor a transaction running at a second terminal.

This mode is started by adding the terminal identifier (termid) of the second terminal to the CEDF command at the first terminal:

```
CEDF tttt
```

The message that CEDF gives in response to this depends on whether there is already a transaction running on the second terminal. If the second terminal is not busy, the message displayed at the first terminal is:

```
TERMINAL tttt: EDF MODE ON
```

and nothing further will happen until a transaction is started on the second terminal, when the PROGRAM INITIATION display will appear.

If a transaction is already running on the second terminal, the message displayed at the first terminal is:

```
TERMINAL tttt: TRANSACTION RUNNING: EDF MODE ON
```

CEDF then picks up control at the next command processed and presents the appropriate display. At the first terminal, you can then follow the sequence of commands in the transaction running at the second terminal.

Monitoring remote transactions

In a multiregion operation (MRO) or an intersystem communication (ISC) environment (APPC only), you can use CEDF (in single screen mode only) for transactions that are defined in the terminal owning region (TOR) as remote. CICS automatically notifies the application owning region (AOR) that the transaction is to be run in EDF mode.

When the remote application ends, if your reply is YES, the terminal remains in EDF mode as usual. However, CICS deletes all the associated temporary storage queues, and none of the previous EDF screens or options is saved: you must type these again for the next transaction. Responding NO at the termination screen ends the EDF session in all participating regions.

Note: CICS supports EDF for remote transactions only when all the communicating regions are using CICS/ESA.

Invoking CEBR and CECI from CEDF

- To invoke the CEBR transaction from the CEDF screen, press PF2. You can then use CEBR commands to browse temporary storage queues and to delete them. CEBR commands are described briefly in Chapter 8, “CEBR - temporary storage browse,” on page 75
- To invoke the CECI transaction from the CEDF screen, press PF5. You can then use CECI commands to display or alter resource status. CECI commands are described briefly in Chapter 10, “CECI - command-level interpreter,” on page 91.

If you are not authorized to use the CEBR or CECI transaction by the external security manager, a warning message will appear on the EDF panel. See the *CICS RACF Security Guide* for information about security checking of transactions running under CEDF.

See Figure 43 for an example of the screen from which CEBR and CECI can be invoked.

```
TRANSACTION: DLID PROGRAM: DLID     TASK: 0000049 APPLID: IYAHZCIB
 ADDRESS: 00000000









WORKING STORAGE IS NOT AVAILABLE
ENTER:  CURRENT DISPLAY
PF1 : UNDEFINED           PF2 : BROWSE TEMP STORAGE PF3 : UNDEFINED
PF4 : EIB DISPLAY         PF5 : INVOKE CECI         PF6 : USER DISP
PF7 : SCROLL BACK         PF8 : SCROLL FORWARD      PF9 : STOP CONDITIONS
PF10: PREVIOUS DISPLAY    PF11: UNDEFINED           PF12: UNDEFINED
```

Figure 43. Typical EDF display from which CEBR or CECI can be invoked

For detailed guidance on using CEBR, see in the *CICS Application Programming Guide*. For detailed guidance on using CECI, see the *CICS Application Programming Guide*.

Using the CEDX transaction

Use CEDX to monitor and debug non-terminal transactions. The transaction you specify for debugging can be:

- Initiated without an associated terminal; for example, by an EXEC CICS START command, or by a transient data queue trigger-level.
- Initiated at a terminal, which can be either the EDF terminal or a different terminal.

CICS intercepts the transaction specified on the CEDX *tranid* command, and displays the EDF diagnostic panels at the terminal at which the EDF command is issued.

CEDX provides the same function and diagnostic display panels as CEDF, and the same basic rules for CEDF also apply to CEDX. However, CICS uses the RESSEC and CMDSEC values of the CEDF transaction to authorize the use of the CEDX transaction.

Command syntax

CEDX

Command options

OFF
specifies that the EDF screen is to be switched OFF. If you specify OFF you must enter the preceding comma, as shown in the following example:

```
CEDX TRNA,OFF
```

ON
specifies that the EDF screen is to be switched ON. The default is ON. If you specify ON you must enter the preceding comma, as shown in the following example:

```
CEDX TRNB,ON
```

tranid
specifies the identifier of a transaction that you want to test using the execution diagnostic facility. The diagnostic panels are displayed on the terminal at which you issue the CEDX *tranid* command.

The transaction you specify on the CEDX command must run in the local CICS region (CRTE followed by CEDX is not allowed).

CEDX cannot interrupt transactions that are already running when you issue the CEDX command. It affects only a transaction that starts *after* you issue the command.

The following diagram illustrates the use of CEDX to invoke EDF for a transaction initiated by an EXEC CICS START command. It shows two terminals. The CICS terminal used for EDF issues a CEDX command before BBBB is started, to invoke EDF services to intercept and debug the specified transaction. The CICS user terminal issues an EXEC CICS START command to start BBBB, without a terminal.

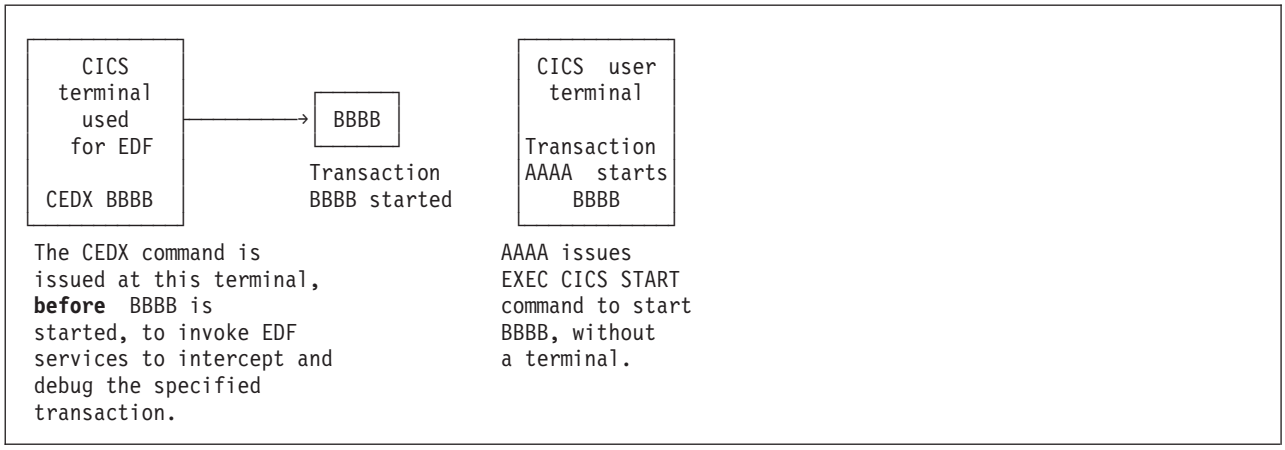


Figure 44. Invoking EDF by CEDX

Chapter 13. CEKL—master terminal (emergency use)

Use CEKL to remove tasks from a system in a situation where you cannot issue the CEMT transaction.

There are two commands, CEKL INQUIRE TASK and CEKL SET TASK. The INQUIRE command displays information about selected tasks. The SET command allows you to remove an individual task using the PURGE, FORCEPURGE, or KILL option.

Note:

1. The CEKL transaction differs from other CICS supplied transactions in that you do not have to install definitions for it.
2. You can start this command at any operating system console that has the authority to issue MODIFY commands for the relevant CICS region. This does not require a CICS terminal definition for the relevant console. You would normally issue the command from an SDSF session (using the ULOG facility to view the extended console output).

The CEKL transaction forms part of the CICS console interface and runs on a dedicated CICS internal TCB which enables commands to be processed even when the rest of CICS is unresponsive.

Minimum abbreviation of keywords

The CEKL command accepts as few characters of a keyword as are needed to uniquely identify it within the request.

In the syntax displayed in this information, the minimum permitted abbreviation is given in uppercase characters, and the rest of the keyword in lowercase characters.

CEKL INQUIRE TASK

INQUIRE TASK returns information about user tasks, and displays the list on the console from which the command was issued and the job log for the CICS region.

Description

From a system console enter:

```
/F cicsregionname,CEKL INQUIRE TASK
```

This displays the current status of the user tasks in your system. The minimum abbreviation is `CEKL I TA`.

From a system console enter: `CEKL INQUIRE TASK` followed by one of the options to limit the range of information that you require. You can restrict the list of tasks using `CEKL I TA D`, the resulting display shows you the details of only those tasks that are marked for dispatch in your system. You can further restrict your selection using the transaction class or transaction id option, for example, `CEKL I TA TRANClass(value)`. This returns a list of the tasks for the specified transaction class.

From a system console enter:

```
/F cicsregionname,CEKL INQUIRE TASK(tasknum)
```

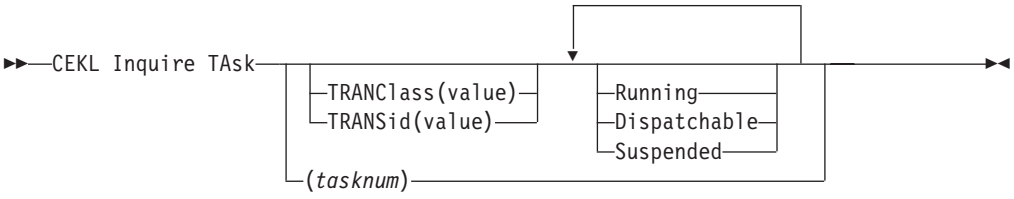
© Copyright IBM Corp. 1982, 2011

This displays the current status of the task in your system. The minimum abbreviation is CEKL I TA(*tasknum*).

Syntax

The syntax of the CEKL INQUIRE TASK command is as follows:

CEKL INQUIRE TASK



Options

Task(*tasknum*)
selects the specified task. *tasknum* is a CICS-generated task number, in the range 1-99999. Task entered without a *tasknum* lists all the tasks on your system. A maximum of 1000 tasks are returned.

TRANClass (*value*)
selects only user tasks from the specified transaction class. TRANClass is a unique transaction class name (up to 8 characters) to which the transaction belongs. You cannot use the symbols * and + to specify transaction class.

TRANSid (*value*)
selects only user tasks with the specified transaction name (up to 4 characters) associated with the transaction. You cannot use the symbols * and + to specify transaction class.

Running
selects tasks that are executing. These tasks are also included if you specify none of the category options (DISPATCHABLE, RUNNING, and SUSPENDED).

Dispatchable
selects tasks that are ready to run (dispatchable). These tasks are also included if you specify none of the category options (DISPATCHABLE, RUNNING, and SUSPENDED).

Suspended
selects tasks that are suspended (tasks waiting for some event or condition). Tasks that have not reached the point of initial dispatch either because the transaction class limit has been reached or because the maxtasks limit has been reached. These tasks are also included if you specify none of the category options (DISPATCHABLE, RUNNING, and SUSPENDED).

Results

A typical output from the inquire command is shown below

```
TAS(nnnnn) ppp ddd cc tttt TCL(tttttttt) USE(aaaaaaaa) ATT(sssss) CPU(sssss)
RUA(sssss) HTI(sssss) HTY(rrrrrrrr) HVA(rrrrrrrr)
```

TAS(*nnnnn*)
nnnnn displays the CICS-generated task number in the range 1–99999 for this task.

ppp *ppp* shows whether the task is being purged, forcepurged or killed. It displays one of the following – PUR, FOR, or KIL. A blank implies that none of the actions have been tried.

ddd *ddd* shows whether the task is in a dispatchable, running or suspended. It displays one of the following – DIS, RUN, or SUS.

cc *cc* specifies the TCB that is associated with this task. The values for *cc* are:

CK indicates that the task is associated with a CICS key open TCB, for example, a JVM (mode J8) TCB.

IN indicates that the task is associated with one of the CICS internal TCBs

QR indicates that the task is associated with the CICS QR TCB

UK indicates that the task is associated with a user key open TCB.

If the task has not reached the point of initial dispatch, the value is blank.

tttt is the name of the transaction (up to 4 characters) associated with the task.

TCL(*tttttttt*)
is the name of the transaction class (up to 8 characters) associated with the task. It the task is not defined in a class then this is blank.

USE(*aaaaaaaa*)
displays the userid for the task.

ATT(*sssss*)
is the length of time (in seconds) since the task was attached. A value of 99999 indicates that the time is greater than 24 hours.

CPUtime(*sssss*)
displays the CPU time, in seconds, used by this task. If monitoring is inactive, the value (*****) is returned. If there is no value returned the task has reached either a transaction class limit or its maxtasks limit. If the task has not reached the point of initial dispatch, the value is blank. A value of 99999 indicates that the time is greater than 24 hours.

RUnAway(*sssss*)
displays the current limit, in seconds, used by CICS to monitor the task for a runaway condition. If the value is zero, CICS is not monitoring the tasks for a runaway condition.

For suspended tasks only, the output includes HTIme(*sssss*), HTYpe(*reason*), and HVAlue(*resourcename*). The values for these three attributes are:

HTIme(*sssss*)
displays the time (in seconds) that the task has been in the current suspended state. A value of 99999 indicates that the time is greater than 24 hours.

HTYpe(*rrrrrrrr*)
displays the reason why the task is suspended.

HVAlue(*rrrrrrrr*)
displays a resource name (up to 16 characters), for example, a file name or a TS queue name.

Note: Blanks may be returned for HTYpe and HVAlue. For example, this occurs if the NAME option is not specified on the WAIT EXTERNAL command.

Sample screen

The sample output shows the output that is generated and returned to the TSO User log. In this example the CEKL command has been used to obtain a list of all the tasks in the system MEACO22X and has returned 14 user tasks. Only the first three tasks are shown in this sample. The first two tasks are suspended and the third one is running.

The output for task 5 indicates that the task is suspended, the TCB is the QR TCB, and the transaction id is XSSY. The userid for the transaction is CICSUSER, the attach time is 6731 seconds, the amount of CPU time used is 39 seconds, and the runaway time limit is 5 seconds. The second line returns the values for the time that the task has been suspended (HTime), the reason why the task is suspended (Htype), and the associated resource name (Hvalue). The second task is similar to the first.

The third task, with the transaction id of L001, is still running in the QR TCB.

```
F MEACO22X,CEKL I TA
+IYK2ZFX3 DFHCQ0243I  CEKL INQUIRE: 00014 task(s) matched selection criteria.
+IYK2ZFX3 TAS(00005)     SUS QR XSSY              USE(CICSUSER) ATT(06731) CPU(00039) RUA(00005)
+IYK2ZFX3 HTI(06732) HTY(ICMIDNTE) HVA(DFHAPTIM)
+IYK2ZFX3 TAS(00006)     SUS QR XSSY              USE(CICSUSER) ATT(06731) CPU(00093) RUA(00005)
+IYK2ZFX3 HTI(03293) HTY(ICEXPIRY) HVA(DFHAPTIX)
+IYK2ZFX3 TAS(00033)     RUN QR LOO1              USE(CICSUSER) ATT(00033) CPU(00023) RUA(00010)
```

Figure 45. CEKL INQUIRE TASK output

The second example shows where a task number has been selected and one record is returned.

```
F MEACO22X,CEKL I TAS(62)
+IYK2ZFX3 DFHCQ0241I  CEKL INQUIRE: task number 00062 found.
+IYK2ZFX3 TAS(00062)     SUS QR XEMT              USE(CICSUSER) ATT(00176) CPU(00069) RUA(00005)
+IYK2ZFX3 HTI(00176) HTY(ZCIOWAIT) HVA(DFHZARQ1)
```

Figure 46. CEKL INQUIRE TASK output

CEKL SET TASK

Requests the cancellation of a user task.

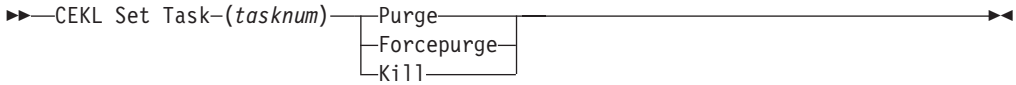
Description

CEKL SET TASK initiates a PURGE, FORCEPURGE, or KILL request.

Syntax

The syntax of the CEKL SET TASK command is as follows:

CEKL SET TASK



Options

Task(*tasknum***)**
The CICS-generated task number. This is in the range 1–99999.

Forcepurge
The task is to be terminated. Data integrity is not guaranteed. Before using forcepurge, you should use purge. In some cases, for example if a task is forcepurged during backout processing, CICS terminates abnormally. If you want to terminate a task but do not want to terminate CICS, you should use PURGE instead of FORCEPURGE.

Kill
The task is to be terminated. System and data integrity is not guaranteed. The KILL option extends the PURGE and FORCEPURGE options. It should be used only after an attempt has been made to PURGE or FORCEPURGE a task. The KILL option does not guarantee integrity of any kind but in some situations it allows the user to free up a stalled region enabling the region to continue processing. In some cases, for example, if a task is killed during backout processing, CICS terminates abnormally.

Purge
The task is terminated. Task termination occurs only when system and data integrity can be maintained.

For each of the preceding options if the task is in any way associated with an open TCB (for example, it involves a Java program running on a J8 or J9 mode TCB), you could experience a short delay before the task is finally purged.

Sample screen

The following figure shows the output that is generated after CEKL has been invoked to initially purge a task, for example `CEKL SET TASK(62) PURGE`.

If you attempt to reissue the purge command the system returns the message informing you that the task is already being purged and it ignores your second command. If you consider that the purge has already taken too long, you can issue a forcepurge command, for example, `CEKL SET TASK(62) FORCEPURGE`.

If the task still remains in the system after an unacceptable length of time you can issue the kill command, for example, `CEKL SET TASK(62) KILL`. This always removes the task from your system. You can confirm this by entering for example,`CEKL INQ TASK(62)`.

```
F MEACO22X,CEKL SET TA(62) PUR
+IYK2ZFX3 DFHCQ0252I  CEKL SET: PURGE request issued for task number 00062.
+IYK2ZFX3 TAS(00062) PUR SUS QR CEMT               USE(CICSUSER) ATT(00270) CPU(*****) RUA(00005)
+IYK2ZFX3 HTI(00271) HTY(ZCIOWAIT) HVA(DFHZARQ1)
F MEACO22X,CEKL SET TA(62) PUR
+IYK2ZFX3 DFHCQ0253I  CEKL SET: PURGE request ignored; task number 00062 is being purged
+IYK2ZFX3 TAS(00062) PUR SUS QR CEMT               USE(CICSUSER) ATT(00288) CPU(*****) RUA(00005)
+IYK2ZFX3 HTI(00289) HTY(ZCIOWAIT) HVA(DFHZARQ1)
+IYK2ZFX3 DFHCQ0252I  CEKL SET: FORCEPURGE request issued for task number 00062.
+IYK2ZFX3 TAS(00062) FOR SUS QR CEMT               USE(CICSUSER) ATT(00270) CPU(*****) RUA(00005)
+IYK2ZFX3 HTI(00271) HTY(ZCIOWAIT) HVA(DFHZARQ1)
+IYK2ZFX3 DFHCQ0257I  CEKL SET: FORCEPURGE request ignored; task number 00062 is being forcepurged
+IYK2ZFX3 TAS(00062) FOR SUS QR CEMT               USE(CICSUSER) ATT(00288) CPU(*****) RUA(00005)
+IYK2ZFX3 HTI(00289) HTY(ZCIOWAIT) HVA(DFHZARQ1)
F MEACO22X,CEKL SET TA(62) KILL
+IYK2ZFX3 DFHCQ0259I  CEKL SET: KILL request issued for task number 00062.
+IYK2ZFX3 TAS(00062) KIL SUS QR CEMT               USE(CICSUSER) ATT(00394) CPU(*****) RUA(00005)
+IYK2ZFX3 HTI(00395) HTY(ZCIOWAIT) HVA(DFHZARQ1)
F MEACO22X,CEKL I TA(62)
+IYK2ZFX3 DFHCQ0242I  CEKL INQUIRE: no tasks matched selection criteria.
```

Figure 47. CEKL SET TASK output

Chapter 14. CEMN - CICS monitoring facility

Use the CEMN monitoring facility transaction to inquire on and set options for the CICS monitoring facility.

CEMN gives you an alternative to the INQUIRE MONITOR and SET MONITOR system programming commands and the equivalent CEMT commands. You can use the transaction to inquire on the settings for the CICS monitoring facility, and to change some of the settings without needing to restart CICS.

The settings which can be changed using CEMN are:

Monitoring Status
Whether monitoring is required.

Exception Class
Whether exception class data is required.

Performance Class
Whether performance class data is required.

Resource Class
Whether resource class data is required.

Compression Status
Whether monitoring data is to be compressed.

Converse Status
Whether separate performance class records are produced for conversational tasks.

Syncpoint Status
Whether separate performance class records are produced for syncpoint requests.

Frequency
The interval at which CICS produces performance class records for long-running tasks. CICS can produce a performance class monitoring record in this way only when the long-running transaction is running on the QR or CO TCBs.

Changing these settings affects the monitoring data which is recorded for tasks that are running at the time you make the change. The effects are the same as if you had changed the settings using the SET MONITOR system programming command or the CEMT SET MONITOR command. “CEMT SET MONITOR” on page 424 explains how data for running tasks is accumulated, recorded, or lost when you change the settings for the CICS monitoring facility.

The settings for which information is displayed, but which cannot be changed using CEMN, are:

File Resource Limit
The maximum number of files for which resource class data is collected.

Tsqueue Resource Limit
The maximum number of temporary storage queues for which resource class data is collected.

Application Naming Status
Whether CICS application naming support is enabled.

© Copyright IBM Corp. 1982, 2011

RMI Status
Whether additional performance monitoring is active for CICS resource managers.

Time Option
Whether the time stamp fields are returned in GMT or local time.

You can change the file resource limit, temporary storage queue resource limit, application naming status, and RMI status using the DFHMCT TYPE=INITIAL macro in the monitoring control table (MCT). You can change the time option using the MNTIME system initialization parameter. A CICS restart is required to implement any of those changes.

Start the CEMN transaction by typing CEMN on the command line of your display and pressing the ENTER key. You get the following display showing the current state of the CICS monitoring facility and the settings of the monitoring options in your own system:

```
CEMN                 CICS Monitoring Control Facility           CJB3 IYK2Z1V3

Type in your choices. When finished, press ENTER.

Item                              Choice       Possible choices

Monitoring Status           ===>  ON           ON, OFf
Exception Class             ===>  ON           ON, OFf
Performance Class           ===>  ON           ON, OFf
Resource Class              ===>  ON           ON, OFf

File Resource Limit         ===>  8            0, 1-64 Files
Tsqueue Resource Limit      ===>  8            0, 1-64 Tsqueues
Application Naming Status   ===>  YES          No, Yes
Compression Status          ===>  YES          No, Yes
Converse Status             ===>  NO           No, Yes
RMI Status                  ===>  YES          No, Yes
Syncpoint Status            ===>  NO           No, Yes
Frequency                   ===>  000000       0, 000100-240000 (hhmmss)
Time Option                 ===>  LOCAL        Gmt, Local



PF1=Help    3=End                                          9=Error List
```

Figure 48. CEMN transaction: initial screen

The input fields can be overtyped with the new values that you require. The File Resource Limit, Tsqueue Resource Limit, Application Naming Status, RMI Status, and Time Option fields are inquire-only fields. When you press ENTER, CEMN issues the necessary commands to set the new values. If there are any errors, PF9 can be pressed to display the error messages. If there is only one short error message, it appears near the bottom of this display.

CEMN is a Category 2 transaction.

Chapter 15. CEMT—master terminal

Use CEMT to invoke all the master terminal functions.

The master terminal program provides dynamic user control of the CICS system. By using this function, an operator can inquire about and change the values of parameters used by CICS, alter the status of the system resources, terminate tasks, and shut down the CICS system.

The following transactions are subsets of CEMT (that is, they contain some but not all of the functions of CEMT):
- CEBT for alternate CICS systems (see Chapter 9, “CEBT - master terminal (alternate CICS),” on page 79)
- CEOT (see Chapter 16, “CEOT - terminal status,” on page 467)
- CEST (see Chapter 19, “CEST—supervisory terminal,” on page 477).

The CECI transaction (see Chapter 10, “CECI - command-level interpreter,” on page 91)) also uses the same command interpreter as CEMT.

You can start CEMT from IBM 3270 family display devices or equivalent terminals or from the system console. Starting CEMT from the operating system console is described in “Using the system console” on page 10. The description of CEMT throughout this section assumes, unless stated otherwise, that you are using a 3270 family display device or equivalent terminal.

To start CEMT, you can either enter the identifier CEMT on the command line by itself, or you can add options on the same line, separated by commas or blanks, to form a complete request.

If you start the transaction by entering CEMT by itself and pressing ENTER, you get a display that prompts you for more information, as described in “Starting the CEMT transaction.”

Starting the CEMT transaction

You can construct a CEMT request in stages, or you can enter a complete request in one go. The CEMT transaction checks the syntax of requests and diagnoses errors. If your request is syntactically correct, it is processed immediately. If your request cannot be processed because of severe syntax errors, the full syntax of your request is displayed.

If you put a question mark (?) in front of your request, the syntax of your request is displayed but it is not processed.

If you put a minus sign (-) immediately following CEMT, and are using the master terminal transaction from the system console, CEMT always displays its results in abbreviated format. This is independent of the number of objects returned on the inquiry. Without the use of a minus sign, CEMT commands issued from the system console display the results in expanded format only if a single object is returned.

After you have initiated the CEMT transaction, you need not reenter CEMT on subsequent requests, because the identifier is implied at the beginning of any further request. However, if you are using the system console, you must always start each master terminal request with CEMT.

© Copyright IBM Corp. 1982, 2011

1. Enter CEMT on the command line of the screen, as follows:
   ```
   CEMT
   ```
2. Press the ENTER key. You get the following response:

```
 STATUS:  ENTER ONE OF THE FOLLOWING

Discard
Inquire
Perform
Set
```

Figure 49. CEMT transaction: initial screen

3. This display prompts you to enter one of the keywords shown.
   a. If you type `D` and press ENTER, you get the following screen:

```
 D
 STATUS:  ENTER ONE OF THE FOLLOWING


Autinstmodel LIBRARY
CONnection   PArtner
CORbaserver  PROCesstype
DB2Conn      PROFile
DB2Entry     PROGram
DB2Tran      Requestmodel
DJar         TCLass
DOctemplate  TCPipservice
Enqmodel     TDqueue
FENode       TErminal
FEPOol       TRansaction
FEPRopset    TSmodel
FETarget     Urimap
FIle
IPconn
JModel
JOurnalname

  S At least 4 characters of DISCARD must be specified.
                                                SYSID=JOHN APPLID=I

PF 1 HELP       3 END       5 VAR                   9 MSG
```

Figure 50. CEMT transaction: DISCARD screen

   b. If you type `I` and press ENTER, you get the following screen:

```
 I
 STATUS:  ENTER ONE OF THE FOLLOWING OR HIT ENTER FOR DEFAULT
AUTInstmodel DSName       JOurnalname  SYDumpcode     Vtam
AUTOinstall  DUmpds       JVM          SYStem         WEb
AUXtrace     ENQ          JVMPool      TAsk           WEBService
BEan         ENQModel     LIBrary      TCLass         WOrkrequest
BRfacility   EXci         LIne         TCPIP
CFdtpool     FEConnection MODename     TCPIPService
CLasscache   FENode       MONitor      TDqueue
CONnection   FEPOol       Netname      TErminal
CORbaserver  FEPRopset    PArtner      TRAnsaction
DB2Conn      FETarget     PIpeline     TRDumpcode
DB2Entry     FIle         PROCesstype  TSModel
DB2Tran      Gtftrace     PROFile      TSPool
DEletshipped Host         PROGram      TSQueue
DIspatcher   INttrace     REquestmodel UOW
DJar         IPconn       RRms         UOWDsnfail
DOctemplate  IRc          STAtistics   UOWLink
DSAs         JModel       STReamname   URimap
```

Figure 51. CEMT transaction: INQUIRE screen

c. If you type `P` and press ENTER, you get the following screen:

```
 P
 STATUS:  ENTER ONE OF THE FOLLOWING

CLasscache
DEletshipped
DJar
DUmp
Endaffinity
Reset
SEcurity
SHUTdown
SNap
STatistics
```

Figure 52. CEMT transaction: PERFORM screen

d. If you type `S` and press ENTER, you get the following screen:

```
  S
  STATUS:  ENTER ONE OF THE FOLLOWING OR HIT ENTER FOR DEFAULT
AUToinstall  FENode       PIpeline         URimap
AUXtrace     FEPool       PROCesstype      Vtam
Brfacility   FETarget     PROGram          WEb
CLasscache   FIle         STatistics       WEBService
CONnection   Gtftrace     SYDumpcode       WOrkrequest
CORbaserver  Host         SYStem
DB2Conn      INttrace     TAsk
DB2Entry     IPconn       TCLass
DB2Tran      IRc          TCPIP
DEletshipped JOurnalname  TCPIPService
DIspatcher   JVM          TDqueue
DOctemplate  JVMPool      TErminal
DSAs         LIBrary      TRAnsaction
DSName       Line         TRDumpcode
DUmpds       MODename     TSqueue
Enqmodel     MONitor      UOW
FEConnection Netname      UOWLink



 PF 1 HELP       3 END       5 VAR                   9 MSG
```

Figure 53. CEMT transaction: SET screen

System parameters

If you press ENTER without selecting one of the CEMT keywords, INQUIRE SYSTEM is assumed and you get a display of the system parameters. Full details are given in “CEMT INQUIRE SYSTEM” on page 274.

Program function (PF) keys

When you start CEMT, the lower part of your display contains a list of PF keys. If your terminal does not have such keys, you can get the same effect by placing the cursor under a name in the list, and pressing ENTER.

Here is what the PF keys do:

PF1 is a general HELP key. It also gives you a list of all the PF keys and what they do.

PF2 switches the display of character values to hexadecimal format on an expanded panel (see “Expanded panels” on page 130).

PF3 ends this master terminal session by terminating the CEMT transaction. If, however, you use this key when you are modifying your display, say by keying in a new command, or by overtyping an old command, this key does not end your session; it is ignored.

PF5 gives a variables display. You can use this to create values for use in a command.

PF7 scrolls backward half a screen.

PF8 scrolls forward half a screen.

PF9 expands messages. If more than one message has been generated in response to your request, a message telling you how many messages have been generated appears near the bottom of the screen. PF9 lets you display all the messages on a separate display. Press ENTER to go back to your original display.

PF10 scrolls backward a full screen.

PF11 scrolls forward a full screen.

Minimum abbreviation of keywords

CEMT accepts as few characters of a keyword as needed to uniquely identify it within the request.

This means that you can enter the keyword TASK as TA or TAS, but you cannot use T, because it can be confused with TCLASS, TDQUEUE, TERMINAL, TRANSACTION, or TRDUMPCODE. You are prompted if the abbreviation you use is not unique. An exception to this rule is SHUTDOWN, for which you must specify at least SHUT.

In the syntax displays on your screen and in the railroad diagrams and syntax descriptions in this book, the minimum permitted abbreviation is given in uppercase characters, with the remainder in lowercase characters.

ALL option

Many of the request formats contain the ALL option. For INQUIRE, this is the default. For example, if you inquire about terminals, you receive information about all terminals unless you specify a terminal identifier, a terminal class identifier, or a system identifier.

ALL is not the default on a SET command, however. If you specify the ALL option on a SET command, any changes you request are made to all resources of the specified type that you are authorized to access.

If there are no resources of the type specified on the SET command, a response of NOT FOUND is given.

CLASS option

When operating on terminals, transactions, and programs, the CLASS option can be used to identify the suffix of a TLT, XLT, or PLT. This limits the scope of the INQUIRE or SET commands to those resources defined in the specified table. Any resources whose definitions have not been installed are displayed as NOT FOUND. For more information, see the *CICS Resource Definition Guide*.

Family of resources (using * and + symbols)

You can use the symbols * and + as part of an identifier to specify a family of resources. * represents any number of characters (including none), and + represents a single character, for example:

A*	all identifiers beginning with A
DATA	all identifiers containing the characters DATA
TERM00+	all 7-character identifiers starting with TERM00.

Use of * and + is restricted to selecting existing resources and cannot be used either to create a new set of resources or to discard resources.

Note: The INQUIRE command and the SET command may use the generic name in a different way. The following two examples illustrate the difference:

```
CEMT INQUIRE FI(*) DSN(ABCD*)
```

lists all the files with data sets with a name beginning with ABCD.

```
CEMT SET FI(*) DSN(ABCD*) CLO
```

associates all your files with data set ABCD* and closes them all.

Thus the generic name has the expected effect only for the resource itself.

List of resource identifiers

You can specify a list of resource identifiers, rather than a single one, separating the items in the list by commas or blanks. Any errors of syntax are reported.

```
CEMT SET TERMINAL(TO1,TO2,TO3,TO4) INSERVICE
```

You cannot use * or + when the identifier is in a list of identifiers.

Note: If you use the comma (,) in the name of a resource identifier, you will not be able to use any commands that permit the use of lists of resources, because the comma will be treated as a list separator, and not as a part of a name.

Subdefining resource groups

When you make an inquiry, you can subdefine a resource group by specifying one or more attributes of the group. For example:

```
CEMT INQUIRE TERMINAL ALL INSERVICE
```

displays the status of all terminals that are in service.

If the subdefined resource group has no members, CEMT returns a NOT FOUND response. For example, if all the files in the system are closed, the command:

```
CEMT INQUIRE FILE ALL OPEN
```

gives the NOT FOUND response.

You cannot subdefine a resource group on the SET command in the same way. For example:

```
CEMT SET TERMINAL ALL INSERVICE PAGE
```

puts all terminals both in service **and** into page status.

However, you can manage such groups of resources by issuing an INQUIRE request, and overtyping the display produced.

Overtyping a display

When you issue an INQUIRE request, you usually get a display that consists of status information for each resource in the specified group. The status information is displayed as a list of abbreviated keywords. You can move the cursor to some fields and change their contents by overtyping. When you press the ENTER key again, CICS reads the contents of all fields that have been changed, and processes any valid operations implied by the changes. If you make an invalid change, you get an error message, and the field is not changed.

When a field is overtyped, CEMT will accept and identify only as few characters as are necessary to uniquely identify the keyword within the valid options for that position. This is usually only one character.

On many of the status displays, the current status is displayed in certain positional fields at the right-hand side of the display. For some fields, the positional display appears only when the status of a particular resource is 'on'. You can overtype the current status (when it is either displayed or blank) to change it.

So, for example, if you type 'CEMT INQUIRE SYD', you get a display of the current settings of the entries in the system dump table. The various attributes of those entries are displayed in fields across the screen. If CICS is to be shut down after the occurrence of an error for this dump code, 'Shu' appears in the third field of attributes for that dump code. If CICS is not to be shut down, however, the 'Shu' space is left blank. If you want CICS to be shut down for this dump code, you can tab to that field and type 'shu' in that blank space in that field only.

The sequence of the positional fields is given in the descriptions of the various CEMT INQUIRE commands.

Whenever you overtype a display, not only is that particular action taken but all the status information is refreshed. You can avoid the overhead of a large number of table searches either by using the SET command, or by limiting your inquiry to a specific number of resources (either by using a 'wild card' in the search argument or by naming a range of resource identifiers).

Tab key

The fields you can change are different in each display. You can detect them, however, by pressing the tab key repeatedly. This causes the cursor to jump from one field to the next.

Scrolling symbol (+ sign)

A plus (+) sign on the first or last line of a display tells you that there is more data above or below your current display. Scrolling backward reveals data above, and scrolling forward reveals data below.

Blank fields in a display

Some displays contain blank fields that allow you to specify options, such as NEWCOPY, on the CEMT SET PROGRAM command, which are not part of the status of a resource. A status, such as NOREAD, which begins NO, is also represented by a blank field. You can detect these fields by using the tab key.

Using the ? key

If you precede the command with ?, the syntax for that command appears on the screen.

For example ?INQUIRE TERMINAL gives the list of options that you can specify with the inquiry command with the default highlighted. See Figure 54 on page 130.

```
?INQ TERM
 STATUS:  COMMAND SYNTAX CHECK
  CEMT  Inquire TErminal()
   < CLass() | ALl >
   < TRansaction() >
   < PRiority() >
   < PAgeable | AUtopageable >
   < Inservice | Outservice >
   < ATi | NOAti >
   < TTi | NOTti >
   < NEtname() >
   < ACquired | RELeased >
   < CReate | NOCreate >
   < REMotesystem() >
```

Figure 54. Sample of screen showing the syntax of a INQUIRE command

Expanded panels

You can select one resource item from a list, then check and change its options if required.

Type the required CEMT INQUIRE or SET command name, then move the cursor to the relevant line on the display panel and press ENTER. This will result in an expanded display showing the attributes of the selected resource item. In the process, an INQUIRE command is executed for the associated resource.

For example, if you enter CEMT INQUIRE TERM, this gives the screen in Figure 55.

```
in task
STATUS:  RESULTS - OVERTYPE TO MODIFY

          *** Sample screen required ***
```

Figure 55. Sample of screen showing a list of terminals

Move the cursor to the left of the first entry and press ENTER. This selects the first terminal, and displays the relevant information for that terminal (see Figure 56 on page 131).

```
in term
RESULT - OVERTYPE TO MODIFY
  Terminal(bell)
  Transaction()
  Priority( 000 )
  Pagestatus( Pageable )
  Servstatus( Inservice )
  Atistatus( Noati )
  Ttistatus( Tti )
  Nature(Local)
  Purgetype(              )
  Netname()
  Termstatus(              )
  Createsess(              )
  Task(00000)
  Remotesystem()
  Rname()
  Rnetname()
  Cid()
```

Figure 56. Terminal settings for a selected terminal

Overtype the displayed settings if you want to make changes. Press ENTER to restore the original panel.

You can view a number of expanded panels in succession by entering '?' or 's' in the space at the beginning of the line for each resource you wish to query. After the first expanded panel, pressing ENTER takes you to the next panel.

Hexadecimal values

Some values such as TSQUEUE and NETUOWID may contain nondisplayable characters, which are shown as periods. You can use PF2 on the expanded panel to see the hexadecimal values of these characters. To switch back to character format, press PF2 again.

Variables

PF5 gives a display of variables, similar to that for the CECI command. Here are some examples of its use.

From a FILE display, you may wish to inquire of a particular DSNAME, which is typically 26 characters and can be up to 44 characters in length. Go to the expanded panel, position the cursor on the DSNAME line, and press PF5. You now have a variable &DSN with the required value. Type I DSN(&DSN) on the command line, and press ENTER.

You may wish to find all the TSQUEUE names beginning with a nondisplayable character - for example, X'FF'. Press PF5 for the variables display. Enter X and 2 to make a variable &X of length 2. Press PF2 to switch to hexadecimal, and overtype '4040' with 'FF5C' ('5C' is hexadecimal for the asterisk (*) symbol). Type I TS(&X) on the command line and press ENTER.

For more information on the variables display, see the *CICS Application Programming Guide*.

Invoking CEMT from an application program

Many of the **CEMT INQUIRE** and SET operations are available through the **EXEC CICS INQUIRE** and **EXEC CICS SET** commands, which you can invoke from an application program; for programming information, see the *CICS System Programming Reference*.

Security checking

The scope of a particular DISCARD, INQUIRE, SET, or PERFORM command can be limited by a combination of several levels of security checking.

The normal transaction-attach security is described in Transaction security the *CICS RACF Security Guide*. CEMT calls the exec interface programs to process the user's commands, in much the same way as CECI or a user's command-level program would. The resource security and the command security used by the command-level programming interface are thus applicable to the master terminal user.

For programming information about command security checking, see the *CICS System Programming Reference*.

CEMT DISCARD commands

The DISCARD command allows you to remove an installed resource definition and its corresponding catalog entry from an active CICS system.

DISCARD does not affect the CICS system definition (CSD) file, and you can therefore reinstate the resource using **CEDA INSTALL**(*resourcename*). DISCARD commands require ALTER access to the resource to provide similar security to that for SET commands (which use UPDATE access). You cannot discard resources that are currently in use or are CICS owned resources (beginning with "DFH" or "C").

You can discard the following resource types:

- Autinstmodel
- Connection (MRO, APPC, or LUTYPE6.1)
- CorbaServer
- DB2 connection
- DB2 entry
- DB2 transaction
- Deployed JAR file
- Document template
- Enqueue model
- FEPI node
- FEPI pool
- FEPI property set
- FEPI target
- File
- IPCONN
- Journal model
- Journal name
- LIBRARY

- Partner
- Pipeline
- Process type
- Profile
- Program
- Request model
- TCP/ IP service
- Temporary storage model
- Terminal
- Transaction
- Transaction class
- Transient data queue
- URIMAP definition
- Web service

Programming information about FEPI commands is in the *CICS Front End Programming Interface User's Guide*.

There are two ways to discard a resource:

1. If you know the name of the resource that you want to discard, type:

   ```
   CEMT DISCARD resourcetype(resourcename)
   ```

2. If you do not know the name of the resource, type:

   ```
   CEMT INQUIRE resourcetype
   ```

   and a screen similar to Figure 57 appears. Use the carriage return key to move the cursor to the beginning of the line for the resource or resources you want to discard, and type “D” as shown and then press ENTER.

```
 IN PROG
 STATUS:  RESULTS - OVERTYPE TO MODIFY
D Prog(ABCDTASK) Len(0002936) Ass Pro Ena Pri    Ced
     Res(000) Use(000000)
  Prog(DFHACP  ) Len(0008232) Ass Pro Ena Pri
     Res(001) Use(000002) Any Cex Ful
  Prog(DFHAKP  ) Len(0005672) Ass Pro Ena Pri
     Res(000) Use(000001) Bel Cex Ful
  Prog(DFHAMP  ) Len(0108432) Ass Pro Ena Pri
     Res(000) Use(000009) Any Cex Ful
  Prog(DFHAPATT) Len(0000632) Ass Pro Ena Sha
     Res(005) Use(000018) Any Cex Ful
  Prog(DFHBRCP ) Len(0000000) Ass Pro Ena Pri
     Res(000) Use(000000) Bel Cex Ful
  Prog(DFHCCNV ) Len(0000000) Ass Pro Ena Pri
     Res(000) Use(000000) Any Cex Ful
D Prog(ERRTASK ) Len(0000448) Ass Pro Ena Pri
     Res(000) Use(000000)
```

Figure 57. Example of screen showing the discarding of resources

Note:

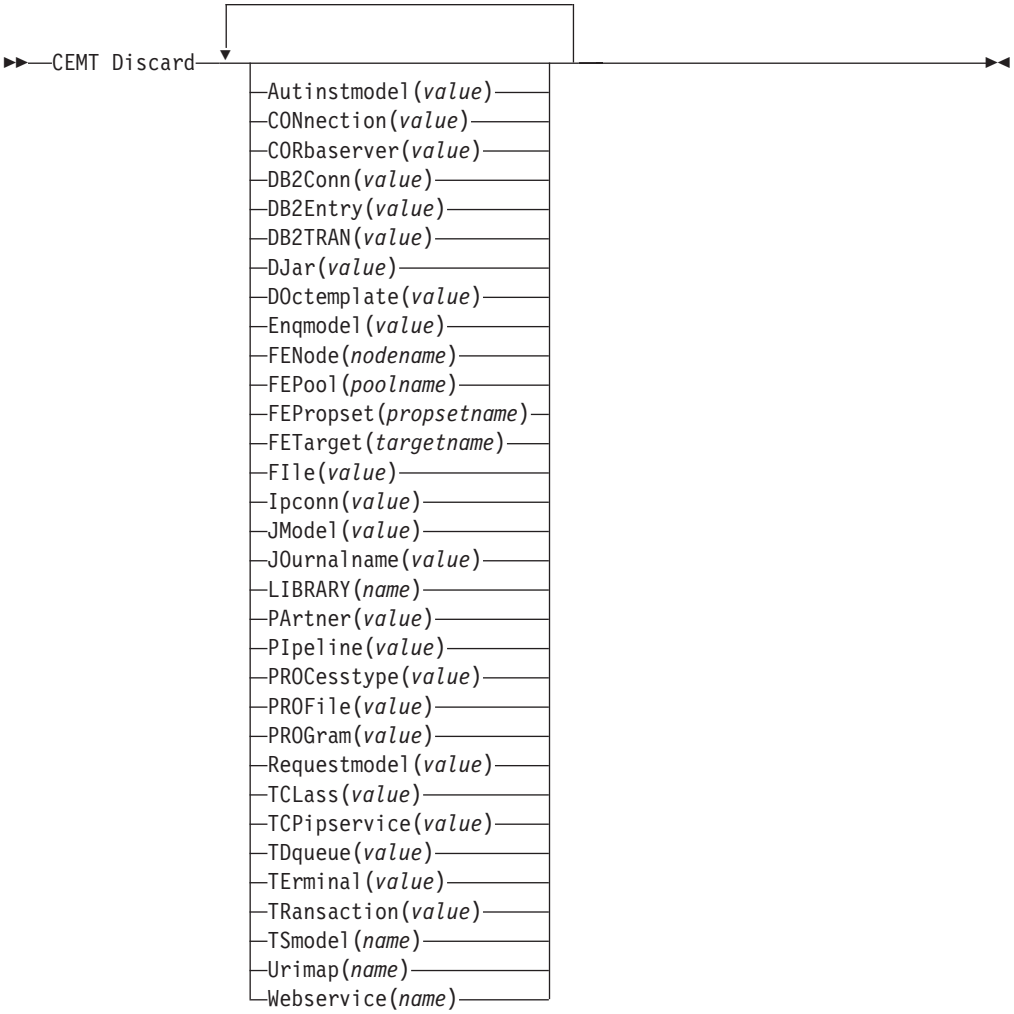
1. A **CEMT INQUIRE** display can be used to discard more than one entry at a time. If you overtype another value as well as typing ‘D’, the ‘D’ is ignored.

2. When discarding a resource, the use of * or + in the resource name is not allowed.

CEMT DISCARD

Syntax

CEMT DISCARD



Options

Autinstmodel(*value***)**
specifies the name of the autoinstall model that is to be removed. The name can be up to 8 characters long. The **DISCARD AUTINSTMODEL** command removes the installed name of an autoinstall model from the autoinstall terminal model (AITM) table and the CICS catalog.

CONnection(*value***)**
specifies the 4-character identifier of the CONNECTION definition to be discarded. When a connection is removed, all its associated sessions are also removed.

For deletion to be successful:

- A non-remote connection must be out of service.

- A remote connection must have no active sessions.
- For MRO connections, the interregion communications facility (IRC) must be closed. (You can use **CEMT SET IRC CLOSED** to close it.)
- If the local system is a member of a VTAM generic resource group, APPC synclevel 2 connections must not have any outstanding recovery information.

  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.

In some unusual circumstances, it is possible for deletion of an LU61 connection to fail, even though the connection is out of service, because its sessions are still in-service. If this happens, set the connection INSERVICE, then OUTSERVICE, then reissue the DISCARD command.

CORbaserver(*value***)**
Specifies the 4-character name of the CORBASERVER definition to be discarded.

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.

DB2Conn(*value***)**
Specifies that the currently installed DB2CONN is to be removed. You can only discard a DB2CONN when the CICS DB2 interface is not active. Note that discarding a DB2CONN also discards all currently installed DB2ENTRYs and DB2TRANs.

DB2Entry(*value***)**
Specifies the name of an installed DB2ENTRY to be removed. A DB2ENTRY can only be discarded when it is in a disabled state.

DB2Tran(*value***)**
Specifies the name of the DB2TRAN to be removed. Transaction IDs referenced by this DB2TRAN will be routed to the pool after the DB2TRAN is removed.

DJar(*value***)**
Specifies the 1–32 character name of the DJAR definition to be discarded.

As well as the DJAR definition, the shelf copy of the deployed JAR file and the beans it contains are also removed from the system. 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.

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 home interfaces of enterprise beans in the deployed JAR file are not retracted from the namespace because they might still be installed in other AORs. 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.

DOctemplate(*value***)**
Specifies the name of the document template to be removed.

Enqmodel(*value***)**
specifies the 8-character identifier of the resource to be discarded.

FENode(*nodename***)**
The name of the FEPI node to be discarded.

FEPOol(*poolname***)**
The name of the FEPI pool to be discarded.

FEPOol(*poolname***)**
The name of the FEPI pool to be discarded.

FEPRopset(*propsetname***)**
The name of the FEPI property set to be discarded.

FETarget(*targetname***)**
The name of the FEPI target to be discarded.

FIle(*value***)**
specifies the name of the file that is to be removed. The name can be up to 8 characters long. The **DISCARD FILE** command removes the installed name of a file from the CICS file control table and the CICS catalog.

Ipconn(*value***)**
specifies the name of the IPCONN to be removed. The name can be up to 8 characters long. You cannot discard an IPCONN unless it is in OUTSERVICE status.

JModel(*value***)**
specifies the name of the installed journal model that you want to remove. The name can be up to 8 characters long. The **DISCARD JMODEL** command removes the journal model name from the installed journal model table and the CICS catalog. It does not affect any existing journals that have been defined using the journal model name. These continue to use their existing definitions unless they are discarded using the **DISCARD JOURNALNAME** command.

JOurnalname(*value***)**
specifies the journal name that you want to remove. The name can be up to 8 characters long. The **DISCARD JOURNALNAME** command removes the journal name from the journal table and the CICS catalog. It does not affect the VSAM data sets currently using the journal as a forward recovery log or autojournal; they continue to use the existing definition until the data sets are closed.

To discard journals defined with a numeric identifier for auto-journaling or forward recovery logs, specify the journal name as DFHJ*nn*, where *nn* is the journal number in the range 1–99.

Note: Specifying DFHJ01 on this command refers to a user journal, *DFHJ01*.

Any future reference to the journal name will cause it to be reinstalled using the journal model resource in effect at that time.

LIBRARY(*name***)**
specifies the name of a LIBRARY resource that you want to remove. The name can be up to 8 characters long. The LIBRARY must be disabled. The **DISCARD LIBRARY** command removes an installed resource definition and its corresponding catalog entry from an active CICS system.

Note: Specifying a LIBRARY name of DFHRPL is invalid, and will result in the message 'NOT VALID FOR RPL' being displayed.

PArtner(*value***)**
specifies the name of the partner that is to be removed. The name can be up to 8 characters long. The **DISCARD PARTNER** command removes the installed name of a partner from the CICS side information table and the CICS catalog.

PIpeline(*value***)**
specifies the name of the PIPELINE that is to be removed. The name can be up to 8 characters long. The **DISCARD PIPELINE** command removes the PIPELINE from the CICS system and the CICS catalog.

PROCesstype(*value***)**
Specifies the 8–character name of the Processtype to be removed. Only disabled processtypes can be discarded. If you are using BTS in a sysplex, it is strongly recommended that you use CPSM to remove them.

PROFile(*value***)**
specifies the name of the profile that is to be removed. The name can be up to 8 characters long. The **DISCARD PROFILE** command removes the installed name of a profile from the CICS profile table (PFT) and the CICS catalog.

PROGram(*value***)**
specifies the name of the program, map set, or partition set that is to be removed. The name can be up to 8 characters long. The DISCARD PROGRAM command removes the installed name of a program, map set, or partition set from the CICS processing program table (PPT) and the CICS catalog.

Requestmodel(*value***)**
specifies the name of the Requestmodel resource definition to be removed.

TCLass(*value***)**
specifies the name of the transaction class that is to be removed. The name can be up to 8 characters long. The DISCARD TCLASS command removes the installed name of a transaction class from the CICS program control table (PCT) and the CICS catalog.

Note: 'TRANCLASS' is also accepted as a synonym for TCLASS.

TCPipservice(*value***)**
Specifies the name of the TCPIP service to be removed.

TDqueue(*value***)**
specifies the name of the transient data queue that is to be removed from a running CICS system. The name can be up to 4 characters long.

Note: When you discard a transient data queue, an implicit DELETEQ command is issued to free the control intervals used by the queue if the queue is an intrapartition queue. If the queue is a logically recoverable intrapartition queue, a syncpoint is taken after the DELETEQ request, but before the queue definition is discarded. This ensures that the associated control intervals have been freed before the definition is discarded.

TErminal(*value***)**
specifies the name of the terminal to be removed. The name can be up to 4 characters long. The **DISCARD TERMINAL** command removes the installed name of a terminal from the terminal control table (TCT) and the CICS catalog.

Only terminals that use VTAM as an access method or consoles are eligible for deletion. Terminals that represent sessions cannot be deleted with **DISCARD TERMINAL**; use the **DISCARD CONNECTION** command instead.

TRansaction(*value***)**
specifies the name of the transaction that is to be removed. The name can be up to 8 characters long. The **DISCARD TRANSACTION** command removes the installed name of a transaction from the CICS program control table (PCT) and the CICS catalog.

TSmodel(*name***)**
specifies the name of the TS model that is to be removed. The name can be up to 8 characters long. You cannot discard a TSmodel that is in use, or one which has a name beginning with DFH.

Urimap(*name***)**
specifies the name of the URIMAP definition that is to be removed. The name can be up to 8 characters long. The **DISCARD URIMAP** command removes the URIMAP definition from the CICS system and the CICS catalog. For a URIMAP definition to be removed, 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 removed.

Webservice(*name***)**
specifies the name of the WEBSERVICE that is to be removed. The name can be up to 8 characters long. The **DISCARD WEBSERVICE** command removes the WEBSERVICE from the CICS system and the CICS catalog.

CEMT INQUIRE commands

Most CEMT requests either inquire about (INQUIRE), or change (SET) the status of one or more named instances of a resource (such as a terminal), a particular subgroup of resources (such as a class of terminals), or all resources of a particular type (such as all terminals).

The INQUIRE command causes the status of the specified resources to be displayed. The SET command makes the changes that you specify, and displays the new status of the resources. No changes are made if there are syntax errors in the SET command.

If, for example, you want to inquire about a resource, enter INQUIRE (or a suitable abbreviation) on the command line. The keywords you can use with CEMT INQUIRE are described in the rest of this section.

If you want to perform those functions that are not concerned with resource status, enter PERFORM (or a suitable abbreviation) on the command line. The keywords you can use with PERFORM are described in “CEMT PERFORM commands” on page 354.

And finally, if you want to change the attributes of a resource, enter SET (or a suitable abbreviation) on the command line. The keywords you can use with SET are described in “CEMT SET commands” on page 370.

You can inquire about any of the options by typing its keyword after INQUIRE on the command line. For example,

```
INQUIRE PROGRAM
```

gives you the status of all programs, and for each program gives its attributes. Full details are given in “CEMT INQUIRE PROGRAM” on page 254.

CEMT INQUIRE AUTINSTMODEL

Find out whether a named autoinstall model is present.

Description

The **INQUIRE AUTINSTMODEL** command returns the names of all the autoinstall models installed in your system. There is no **CEMT SET AUTINSTMODEL** function, but the **INQUIRE AUTINSTMODEL** screens allow you to use the DISCARD function.

For information about autoinstall models, see the *CICS Resource Definition Guide*.

Input

Press the Clear key and type `CEMT INQUIRE AUTINSTMODEL` (the minimum abbreviation is `CEMT I AUTI`).

You will get a display screen, a sample of which is shown below.

If you do not type the name of a specific autoinstall model (the name can be up to 8 characters long—for example, `cemt i auti(dfhlu0e2)`), the names of all the autoinstall models will be displayed.

Sample screen

```
 IN AUTIN
 STATUS:  RESULTS
  Auti(DFHLU0E2)
  Auti(DFHLU0M2)
  Auti(DFHLU0M3)
  Auti(DFHLU0M4)
  Auti(DFHLU0M5)
  Auti(DFHLU2  )
  Auti(DFHLU2E2)
  Auti(DFHLU2E3)
  Auti(DFHLU2E4)
  Auti(DFHLU2M2)
  Auti(DFHLU2M3)
  Auti(DFHLU2M4)
  Auti(DFHLU2M5)
  Auti(DFHLU3  )
  Auti(DFHLU62T)
  Auti(DFHSCSP )
  Auti(DFH3270 )
+ Auti(DFH3270P)
```

Figure 58. CEMT INQUIRE AUTINSTMODEL screen

Note: The ‘+’ symbol in the left margin by the last item on the screen indicates that further model names are displayed when the PF8 key is pressed.

CEMT INQUIRE AUTOINSMODEL

Displayed fields

All
Automatic switching occurs as necessary until the end of this CICS session without the need for operator intervention.

CEMT INQUIRE AUTOINSTALL

Retrieve information about autoinstall values.

Description

The **INQUIRE AUTOINSTALL** command displays information and parameters that control the automatic installation (autoinstall) of VTAM terminals, APPC connections, virtual terminals including bridge facilities used by the 3270 bridge mechanism and MVS consoles.

For background information about autoinstall, see the *CICS Resource Definition Guide*.

Input

Press the Clear key and type `CEMT INQUIRE AUTOINSTALL` (the minimum abbreviation is `CEMT I AUTO`).

You will see a display screen, a sample of which is shown below. If you want to change any of the values shown, you can tab to the highlighted fields and overtype them as required.

Sample screen

```
INQ AUTOINSTALL
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Auto Max( 100 ) Pro( DFHZATDX ) Cur(000) Pro Ena
```

Figure 59. CEMT INQUIRE AUTOINSTALL screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 60 on page 141.

```
IN AUTOIN
RESULT - OVERTYPE TO MODIFY
  Autoinstall
  Maxreqs( 100 )
  Program( DFHZATDX )
  Curreqs( 000 )
  Consoles( Fullauto )
  Enablestatus( Enabled )
  Aibridge( Autotermid )
```

Figure 60. The expanded display of an individual entry

CEMT INQUIRE AUTOINSTALL

Displayed fields

Aibridge
indicates whether the autoinstall user replaceable module (URM) is called for bridge facilities. The values are:

AUTOTERMID
Bridge facilities are defined automatically by CICS. The autoinstall URM is not called.

URMTERMID
The autoinstall URM is called.

Autoinstall
identifies this panel as relating to terminal autoinstall.

Consoles(*value***)**
displays the status of autoinstall for MVS consoles in the CICS region. The values are:

Fullauto
Autoinstall for consoles is active without any support from the autoinstall control program. CICS generates the required terminal ID automatically without calling the control program, and sets the default delete-delay period (60 minutes) for the console entry.

Progauto
Autoinstall for consoles is active, with the support of the autoinstall control program (specified on the **AIEXIT** system initialization parameter). The autoinstall control program is required to provide the terminal ID and, optionally, a delete-delay period.

Noauto
Autoinstall for consoles is not active.

Note: You can change this attribute by overtyping it with a different value.

Curreqs(*value***)**
displays the number of autoinstall requests that are currently being processed. (This is not the number of terminals currently installed using autoinstall.)

Enablesatus(*value*)
displays the overall status of the terminal autoinstall facility, indicating whether CICS is enabled to install VTAM terminals and MVS consoles. The values displayed are:

Enabled
CICS can currently install consoles, or terminals, or both. To determine which of these the facility is enabled for, check the values returned on the MAXREQS and CONSOLES options. Enabled is returned for the following conditions:

Terminals
MAXREQS not equal 0 and autoinstall control program is enabled.

Consoles
1. CONSOLES displays FULLAUTO.
2. CONSOLES displays PROGAUTO and the autoinstall control program is enabled.

Disabled
Neither consoles nor terminals can be autoinstalled in CICS. DISABLED is returned for the following conditions:

Terminals
MAXREQS equal 0, or autoinstall control program is disabled.

Consoles
1. CONSOLES displays NOAUTO.
2. CONSOLES displays PROGAUTO but the autoinstall control program is disabled.

Maximum(*value*)
displays the largest number of autoinstall requests that are allowed to queue at any one time. (This is not the largest number of terminals that can be installed using autoinstall.)

Note: You can reset this value by overtyping it with a different value.

Program(*value*)
displays the name of the program that controls the autoinstall process. This is either the CICS-supplied default autoinstall program, DFHZATDX, or a user-written program.

Note: You can reset this value by overtyping it with a different value.

CEMT INQUIRE AUXTRACE

Function

Retrieve information about the status of auxiliary trace.

Description

Auxiliary trace entries are made to a pair of CICS-controlled BSAM data sets, 'A' and 'B'.

INQUIRE AUXTRACE tells you:

- Whether the auxiliary trace data set is open or closed
- Which of the auxiliary trace data sets is active, Cur(A) or Cur(B)
- Whether an auxiliary trace is in progress (started) or not (stopped).

You can start, stop, or pause tracing, and specify that automatic switching of the auxiliary data set is to occur when needed. You can use the SWITCH option to change the data set.

Instead of using the SET command, you can overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 128).

For information on using auxiliary tracing, see the *CICS Problem Determination Guide*. See also the description of the CETR transaction, Chapter 20, “CETR - trace control,” on page 479.

Input

Press the Clear key and type `CEMT INQUIRE AUXTRACE` (the minimum abbreviation is `CEMT I AUX`).

You will get a display screen, a sample of which is shown below. If you want to change any of the values shown, you can then tab to the highlighted or blank fields and overtype them as required.

Sample screen

```
I AUX
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Aux Cur(A) Sto
```

Figure 61. CEMT INQUIRE AUXTRACE screen

Note: There are blank fields on the screen where a value does not apply or is ‘negative’; that is, the value begins with ‘No’. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 62.

```
I AUX
RESULT - OVERTYPE TO MODIFY
  Auxtrace
  Curauxds(A)
  Status( Stop )
  Switchstatus( Noswitch )
  Switchaction(             )
```

Figure 62. The expanded display of an individual entry

CEMT INQUIRE AUXTRACE

Displayed fields

Auxiliary
identifies this panel as relating to auxiliary trace.

Curauxds(*value***)**
displays a 1-character identifier of the current auxiliary data set, which can be 'A' or 'B'. This value is blank if there is no auxiliary trace data set.

Status(*value***)**
displays the status of auxiliary tracing in your CICS system. The values are:

Start Auxiliary tracing is in progress (the auxiliary trace data set is open).

Pause
Auxiliary tracing has stopped, but the trace data set has been left open. A subsequent START request causes trace entries to be written immediately following those that were written before the PAUSE request.

Stop Auxiliary tracing has stopped, and the trace data set is closed.

Note: You can reset this value by overtyping it with a different value.

Switchaction(*value) (input only field***)**
specifies that the trace data set is to be switched. The value is:

Switch
Makes an immediate switch from the current auxiliary trace data set to the alternate trace data set.

Switchstatus(*value***)**
displays whether automatic data set switching is to occur when the current auxiliary trace data set becomes full. The values are:

Noswitch
Switching does not occur without operator intervention.

NExt Automatic switching occurs when the current trace data set is full, but not subsequently.

ALl Automatic switching occurs as necessary until the end of this CICS session without the need for operator intervention.

Note: You can reset this value by overtyping it with a different value.

CEMT INQUIRE BEAN

Retrieve information about an installed enterprise bean.

Description

INQUIRE BEAN returns information about any currently installed enterprise bean.

Input

Press the Clear key and type `CEMT INQUIRE BEAN` (the minimum abbreviation is `CEMT I B`). You get a display that lists the names and status of any installed enterprise bean. Note that you cannot change any of the displayed information.

Sample screen

```
INQ BEAN
STATUS:  RESULTS
 Bean(Managed                  ) Corba(CORA) Djar(IJAR                          )
 Bean(Mandatory                ) Corba(CORB) Djar(BJAR                          )
 Bean(MasterTs                 ) Corba(CORA) Djar(IJAR                          )
 Bean(Never                    ) Corba(CORB) Djar(BJAR                          )
 Bean(NotSupported             ) Corba(CORA) Djar(IJAR                          )
 Bean(Required                 ) Corba(CORB) Djar(BJAR                          )
 Bean(RequiresNew              ) Corba(CORA) Djar(IJAR                          )
 Bean(Supports                 ) Corba(CORB) Djar(BJAR                          )









  SYSID=QT20 APPLID=IYCQST20
RESPONSE: NORMAL                            TIME:  11.09.09  DATE: 19.01.01
  PF 1 HELP       3 END       5 VAR       7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 63. CEMT INQUIRE BEAN screen

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format.

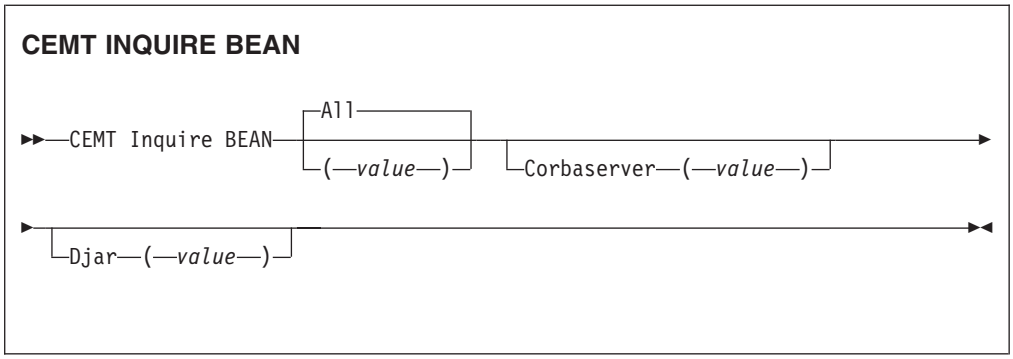
```
i bean
RESULT
  Bean(Managed)
  Corbaserver(CORA)
  Djar(IJAR                          )




SYSID=QT20 APPLID=IYCQST20
TIME:  11.10.42  DATE: 19.01.01
  PF 1 HELP 2 HEX 3 END       5 VAR       7 SBH 8 SFH       10 SB 11 SF
```

Figure 64. The expanded display of an individual entry

If you inquire against a specific bean and press ENTER, CICS displays an expanded format.

```
inq bean(Managed) corba(CORA)
STATUS:  RESULTS
 Bean(Managed                ) Corba(CORA)
 Djar(IJAR                        )




SYSID=QT20 APPLID=IYCQST20
TIME:  11.10.42  DATE: 19.01.01
  PF 1 HELP 2 HEX 3 END       5 VAR       7 SBH 8 SFH       10 SB 11 SF
```

Figure 65. The expanded display of a specific entry



Displayed fields

All
is the default.

(*value***)**
is a 1–240 character bean name.

Corbaserver (*value***)**
displays a 4-character CorbaServer name.

Djar (*value***)**
displays a 1–32 character deployed JAR file name.

CEMT INQUIRE BRFACILITY

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

Description

INQUIRE BRFACILITY 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.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE BRFACILITY` (the minimum abbreviation is `CEMT I BR`). You get a display that lists the status of any currently allocated bridge facilities.
- Type `CEMT INQUIRE BRFACILITY` (`CEMT I BR`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `CEMT I BR S`, the resulting display will show you the details of only those bridge facilities that were allocated for Link3270 requests (SHARED). To find out which options you can enter on the `CEMT I BR` request, you can enter `?CEMT I BR`, for example.

```
?CEMT I BR
 STATUS:  COMMAND SYNTAX CHECK
  CEMT  Inquire BRfacility()
   < ALl >
   < Netname() >
   < TErmid() >
   < TRansaction() >
   < TAskid() >
   < Userid() >
   < Keeptime() >
   < RELeased >
   < ACquired | AVailable >
   < LINKSYSTem() >
   < LINKSYSNet() >
   < REMOTESYSTem() >
   < REMOTESYSNet() >
   < LOcal | Shared >
```

Figure 66. CEMT INQUIRE BRFACILITY syntax screen

To change various attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see "Overtyping a display" on page 128).
- Use the CEMT SET BRFACILITY command.

Sample screen

```
  I BR
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Brf(0008000100000016) Net(AAA.    ) Ter(AAA.) Tra(NACT)
     Tas(0005730) Use(JANEDJ  ) Kee(97105) Acq
```

```
                                                SYSID=ST50 APPLID=IYCSST50
  RESPONSE: NORMAL                          TIME:  11.13.29  DATE: 08.05.01
PF 1 HELP       3 END       5 VAR       7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

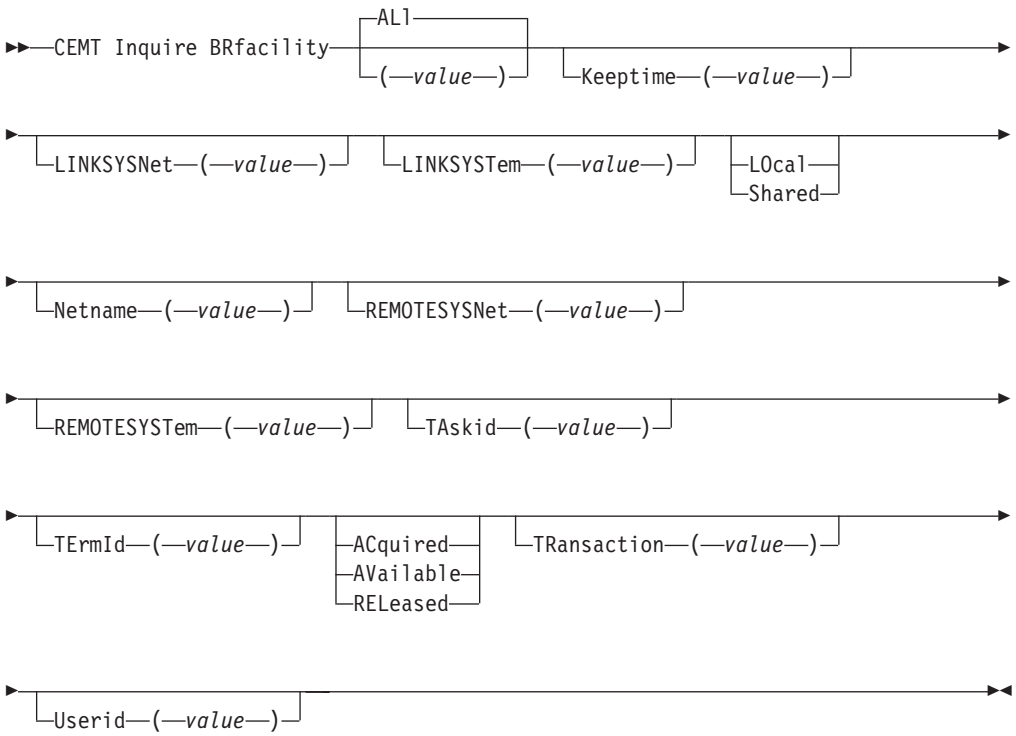
Figure 67. CEMT INQUIRE BRFACILITY screen

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown:

```
I BR
RESULT - OVERTYPE TO MODIFY
  Brfacility(0008000100000016)
  Netname(AAA.)
  Termid(AAA.)
  Transaction(NACT)
  Taskid(0005730)
  Userid(JANEDJ)
  Keeptime(96908)
  Termstatus( Acquired )
  Linksystem()
  Linksysnet()
  Remotesystem(CTGC)
  Remotesysnet( DFHGEN)
  Namespace(Shared)




                                          SYSID=ST50 APPLID=IYCSST50
                                       TIME:  11.16.46  DATE: 08.05.01
PF 1 HELP 2 HEX 3 END       5 VAR       7 SBH 8 SFH       10 SB 11 SF
```

Figure 68. The expanded display of an individual bridge facility entry

CEMT INQUIRE BRFACILITY



Displayed fields

(*value***)**
is the 8–byte bridge facility token of a specific bridge facility to be displayed.

All
is the default. All bridge facilities are displayed.

Keeptime *(value)*
displays 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 keeptime specified when the facility was allocated,
  - or a default value of 5 minutes if no keeptime was specified,

  if the keeptime is larger than the value of SIT parameter **BRMAXKEEPTIME**, it is reduced to BRMAXKEEPTIME.
- If the facility being displayed is a Web-Bridge, the keeptime is initially set as the Web delay terminal keep time (the second part of the **WEBDELAY** SIT parm).

Linksysnet(*value***)**
displays 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 router, 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 router region.

Linksystem(*value***)**
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 router, 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 router region.

Namespace
displays the scope of the namespace used to allocate bridge facility names. 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 router regions in the CICSplex who have access to a shared DFHBRNSF namespace file.

Netname(*value***)**
displays the 8-byte virtual netname name of the bridge facility about which you are inquiring.

Remotesysnet(*value***)**
displays an 8–byte field giving the applid of the bridge router region. This field is only set in the AOR region. It is blank if the request is processed in the router region and not sent to an AOR.

Remotesystem(*value***)**
displays an 4–byte field giving the sysid of the bridge router region. This field is only set in the AOR region. It is blank if the request is processed in the router region and not sent to an AOR.

Taskid(*value***)**
displays 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(*value*)
displays the 4-byte virtual terminal name of the bridge facility about which you are inquiring.

Termstatus
displays the status of the bridge facility. Values are:

ACQUIRED
The bridge facility is currently in use.

AVAILABLE
The bridge facility is not in use. It can be reused by the client.

RELEASED
SET BRFACILITY RELEASED has been issued for the bridge facility. It will be deleted on the next cleanup cycle.

Transaction(*value*)
displays 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(*value*)
displays an 8–byte field giving the userid associated with this bridge facility.

CEMT INQUIRE CFDTPOOL

Retrieves information about coupling facility data table pools.

Description

INQUIRE CFDTPOOL returns information about any currently installed CFDT pool names. CICS installs these automatically whenever it installs a file resource definition that refers to a CFDT.

Input

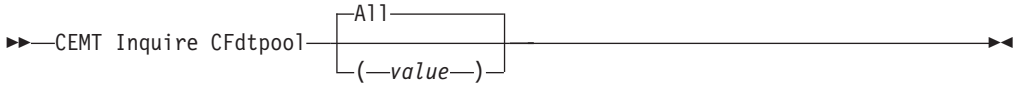
Press the Clear key to clear the screen, and type `CEMT INQUIRE CFDTPOOL` (the minimum abbreviation is `CEMT I CF`). You get a display that lists the names and status of any coupling facility data table pools to which CICS is connected through a CFDT server. Note that you cannot change any of the displayed information.

Sample screen

```
inquire cfdt
STATUS:  RESULTS
 Cfd(DTPOOL1 ) Con









                                        SYSID=HA61 APPLID=CICSHA61
  RESPONSE: NORMAL                     TIME:  15.54.22  DATE: 09.03.98
PF 1 HELP       3 END       5 VAR       7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 69. CEMT INQUIRE CFDTPOOL screen

CEMT INQUIRE CFDTPOOL



Displayed fields

All
This is the default. CICS displays the names of all installed CFDT pools.

Cfdtpool(*value*)
displays information about a named CFDT pool.

CICS creates a resource entry for a CFDT pool when a file resource definition that refers to a coupling facility data table is first installed. The resource entry is cataloged, and reinstalled from the catalog on a warm or emergency restart.

Note: You cannot discard CFDT pool entries other than by a cold or initial start of the CICS region.

Connstatus
displays the connection status for the named CFDT pool. The values are:
Connected
The server for the coupling facility data table pool is available in this MVS image, and CICS is currently connected to it.
Unconnected
The server for the coupling facility data table pool is available in this MVS image, but CICS is not currently connected to it.
Unavailable
The server for the coupling facility data table pool is currently unavailable in the MVS image.

CEMT INQUIRE CLASSCACHE

Retrieves information about the shared class cache.

Description

INQUIRE CLASSCACHE returns information about the shared class cache in the CICS region, and reports the presence of any old shared class caches that are awaiting deletion.

Input

Press the Clear key to clear the screen, and type `CEMT INQUIRE CLASSCACHE` (the minimum abbreviation is `CEMT I CL`). You get a display that describes the active shared class cache and indicates any old shared class caches, a sample of which is shown below.

You can change the status of autostart (to Enabled or Disabled) by overtyping the value for Autostartst. You can also change the displayed values by using CEMT commands as follows:

- Change the status of autostart (Autostartst) by using the CEMT SET CLASSCACHE command, or by using the Enabled or Disabled option when you are performing the Phaseout, Purge or Forcepurge operation with the CEMT PERFORM CLASSCACHE command.

- Change the size of the shared class cache (Cachesize), by using the Cachesize option when you are performing the Start or Reload operation with the CEMT PERFORM CLASSCACHE command. The new size that you specify is used for the new shared class cache.
- If you are using the IBM SDK for z/OS, V1.4.2 to provide Java support, change the JVM profile that is used for the master JVM that initializes the shared class cache (Profile), by using the Profile option when you are performing the Start or Reload operation with the CEMT PERFORM CLASSCACHE command. The new JVM profile that you specify is used for the master JVM that starts the new shared class cache.

Sample screen

```
i classcache
RESULT - OVERTYPE TO MODIFY
  Classcache
  Autostartst( Enabled )
  Status(Started)
  Profile(DFHJVMCC)
  Datestarted(10/14/05)
  Timestarted(16:57:24)
  Totaljvms(0002)
  Oldcaches(0000)
  Phasingout(0000)
  Reusest(Reuse)
  Cachesize(50M                )
  Cachefree(45260539           )
```

Figure 70. CEMT INQUIRE CLASSCACHE screen

INQUIRE CLASSCACHE

Displayed fields

Autostartst
displays the status of autostart for the shared class cache. The values are:

Enabled

When autostart is 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

When autostart is disabled, if the shared class cache has been stopped or has not yet been started on this CICS execution, an explicit PERFORM CLASSCACHE command is required to start it. If the status of the shared class cache is STOPPED and autostart is disabled, and CICS receives a request to run a Java application in a JVM whose profile requires the use of the shared class cache, the application cannot run.

Cachefree(*value***)**
displays the amount of free space in the shared class cache, in bytes.

CACHEFREE only gives a valid result when you are using the IBM SDK for z/OS, V1.4.2 to provide Java support. If you are using the IBM SDK for z/OS, V5, the value is always the same as CACHESIZE.

Cachesize(*value***)**
displays the size of the shared class cache, in bytes, kilobytes (K), megabytes (M) or gigabytes (G).

- If the status of the shared class cache is STOPPED, this is 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 is the size of the current shared class cache.
- If the status of the shared class cache is RELOADING, this is the size of the new shared class cache that is being loaded.

Datestarted(*mm/dd/yy, dd/mm/yy, or yy/mm/dd***)**
displays the date on which the current shared class cache was started. The format of the date depends on the value that you selected for the DATFORM system intialization parameter for your CICS region.

Oldcaches(*value***)**
displays the number of old shared class caches that are still present in the region because they are waiting for JVMs that are using them to be phased out. If the status of the current shared class cache is STOPPED, and JVMs are still using it, then that shared class cache is included in the number of old shared class caches.

Phasingout(*value***)**
displays the number of JVMs that are using an old shared class cache, and are being phased out. If the status of the current shared class cache is STOPPED, then any JVMs that are still using it are included in the number of JVMs being phased out.

Profile(*value***)**
displays the name of the JVM profile for the master JVM.

If you are using the IBM SDK for z/OS, V5 to provide Java support, the shared class cache is not initialized by a master JVM, so this field is blank. When you are using the IBM SDK for z/OS, V1.4.2:

- If the status of the shared class cache is STOPPED, this is the 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 is the 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 reload.

The displayed JVM profile is used for the master JVM 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
returns a CVDA value indicating the reuse status of the JVMs that use the shared class cache. With the IBM SDK for z/OS, V1.4.2, this is also the reuse status of the master JVM that initializes the shared class cache.

REUSE
The JVMs that use the shared class cache, and if applicable, the master JVM are continuous. In this CICS release, JVMs associated with the shared class cache are always continuous (reusable) JVMs.

UNKNOWN
The shared class cache is not started. Although CICS cannot identify the reuse status in this situation, the status always becomes REUSE when the shared class cache is started.

Status
displays the status of the current shared class cache. The values are:

Started
The shared class cache is ready, and it can be used by JVMs. This value in the CEMT display includes both the status STARTED and the transient status RELOADING, which occurs when a new shared class cache is being loaded to replace the existing shared class cache. While the shared class cache is reloading, JVMs that are using the shared class cache, both those that were already allocated to tasks and those that were allocated to tasks after the command was issued, continue to use the existing shared class cache until the new shared class cache is ready.

Stopped
The shared class cache has either not been initialized on this CICS execution, or it has been stopped by a PERFORM CLASSCACHE command or a SET JVMPOOL command. If autostart is disabled, requests to run a Java application in a JVM whose profile requires the use of the shared class cache 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. This value in the CEMT display includes both the status STOPPED and the transient status STARTING, which occurs when the shared class cache is being initialized. While the shared class cache is starting, new JVMs that require the use of the shared class cache wait until the startup process is complete and the shared class cache is ready. If initialization of the shared class cache is unsuccessful, any waiting requests for JVMs that require the use of the shared class cache fail.

Timestarted(*hh:mm:ss*)
displays the time, as an absolute value measured from midnight, that the current shared class cache was started. The time is in the format `hh:mm:ss`.

Totaljvms(*value*)
displays the number of JVMs in the CICS region that are using a shared class cache. This includes both the JVMs that are using the current shared class cache, and any JVMs that are using an old shared class cache and are being phased out.

CEMT INQUIRE CONNECTION

Function

Retrieve information about an MRO or ISC over SNA connection to a remote system.

Description

CEMT INQUIRE CONNECTION returns information about a connection from your local CICS region to another CICS region or another system.

Note: CEMT INQUIRE CONNECTION returns information about MRO and ISC over SNA connections. See also "CEMT INQUIRE IPCONN" on page 219. The CEMT INQUIRE IPCONN command returns information about IPIC connections (also known as "*IP connections*").

For information about the different kinds of intercommunication connections, see the *CICS Intercommunication Guide*.

You can reset the options either by typing the SET command or by overtyping at the appropriate place on the INQUIRE screen.

If you want to install a new connection definition when one is already installed, and:

- You are using APPC,
- You are using MRO,
- You are using remote,
- You are using LU6.1, you must set the connection OUTSERVICE and, if it is VTAM-connected, release it before using the CEDA INSTALL commands for your new definition.

See the *CICS Resource Definition Guide* for further information about connections.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE CONNECTION` (the minimum abbreviation is `CEMT I C`). You get a display that lists the current status.
- Type `CEMT INQUIRE CONNECTION` (`CEMT I C`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i c ins acq`, the resulting display will show you the details of only those connections that are in service and acquired.

You can then tab to the highlighted or blank fields and overtype them with the required values.

(*value***)**
specifies one or more names (1–4 characters) defined for an interregion communication (IRC) or an intersystem communication (ISC) connection.

All
is the default.

Sample screen

```
IN CONN
STATUS:  RESULTS - OVERTYPE TO MODIFY
  Con(G1GA)                   Ins     Irc Exci
  Con(ISC1) Net(AUHZCIC1)     Ins Acq Vta Appc     Xok Unk
                                          Nqn(GBIBMIYA.AUHZCIC1)
  Con(MRO2) Net(CICSHAA1)     Ins Acq Irc              Unk
  Con(MRO4) Net(CICSHAA2)     Ins Acq Irc              Unk
  Con(MRO5) Net(CICSHAA3)     Ins Acq Irc              Unk
  Con(MRO6) Net(CICSHAB1)     Ins Acq Xcf
  Con(VTA1) Net(IYCQTC03)     Ins Acq Vta Appc
                                          Nqn(GBIBMIYA.IYCQTC03)
```

Figure 71. CEMT INQUIRE CONNECTION screen

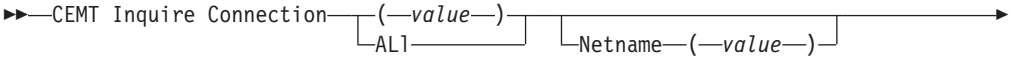
Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

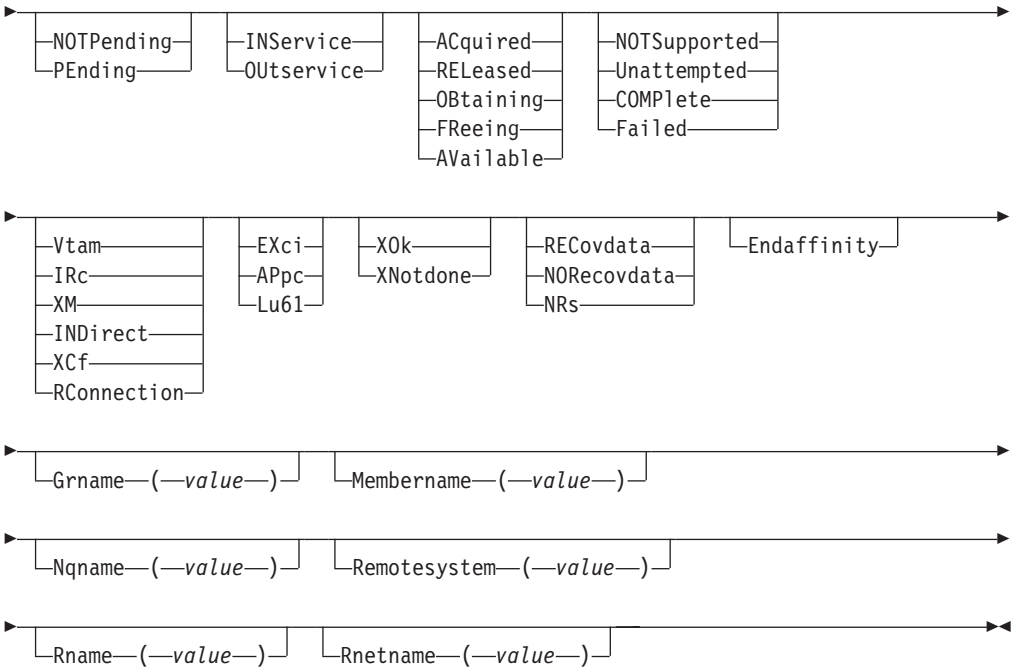
If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 72.

```
IN CONN
RESULT - OVERTYPE TO MODIFY
  Connection(MRO5)
  Netname(CICSHAA3)
  Pendstatus( Notpending )
  Servstatus( Inservice )
  Connstatus( Acquired )
  Accessmethod(Irc)
  Purgetype(            )
  Protocol()
  Xlnstatus()
  Recovstatus( Norecovdata )
  Uowaction(             )
  Cqp( Notsupported )
  Grname()
  Membername()
  Affinity(              )
  Nqname()
  Remotesystem()
  Rname()
  Rnetname()
```

Figure 72. The expanded display of an individual entry

CEMT INQUIRE CONNECTION





Displayed fields

Accessmethod
displays the access method in use for this connection. The values are:

Vtam The connection is used for intersystem communication (ISC).

Irc The connection is used for multiregion operation (MRO), and has been defined to use the interregion communication (IRC) program for communication. If the CONNECTION is ACQUIRED, the MRO partner is running on the same MVS image. If the CONNECTION is RELEASED, the MRO partner may not be on the same MVS image; if it is not, the XCF access method will be used when the connection becomes ACQUIRED.

Xm The connection is used for multiregion operation (MRO) and has been defined to use MVS cross-memory (XM) services for communication. If the CONNECTION is ACQUIRED, the MRO partner is running on the same MVS image. If the CONNECTION is RELEASED, the MRO partner may not be on the same MVS image; if it is not, the XCF access method will be used when the connection becomes ACQUIRED.

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.

Xcf The connection is used for multiregion operation (MRO), and communication uses the cross-system coupling facility (XCF) of MVS. XCF is used for MRO links between CICS regions on different MVS images within an z/OS sysplex. It is selected dynamically by CICS for such links when the access method is defined as IRC or XM in the connection definition.

Rconnection
The connection is remote—that is, it is owned by another CICS system.

Affinity (*input only field***)**
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 LU61 connections. The connection must be out of service and in NORECOVDATA state. The value is:

Endaffinity
VTAM is to end an affinity owned by CICS.

Connection(*value*)
identifies this panel as relating to system connections, and displays the 4-character name by which the connection is known.

Connstatus
displays the state of the connection between CICS and a remote system. The remote system can be an APPC or an MRO partner identified by (*value*) on the CEMT INQUIRE CONNECTION command. The ACQUIRED and RELEASED values are common to both APPC and MRO; the others are unique to APPC. The values are:

Acquired (APPC and MRO)
The CONNECTION is acquired. The criteria for ACQUIRED for VTAM links are:
- The partner LU has been contacted.
- Initial 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 the CICS interregion communication program (DFHIRP).
- A connection request by each side has been successful for at least one session, and therefore each side can send and receive data.

Released (APPC and MRO)
The connection is released. Although the connection might be in service, it is not usable.

In the case of an MRO link, the released status can be caused by any one of a number of conditions. For example, it could be because the CICS region on the other side has not yet initialized, or not yet signed on to the CICS interregion communication program (DFHIRP); or it could be because CICS interregion communication may have been closed on the other side, or the connection on the other side may have been set out of service.

Obtaining (APPC only)
The connection is being acquired. The connection remains in the OBTAINING state until all the criteria for ACQUIRED have been met.

Freeing (APPC only)
The connection is being released.

Available (APPC only)
The connection is acquired but there are currently no bound sessions because they were unbound for limited resource reasons.

Connstatus can be reset with Acquired or Released.

Cqp(*value*)
displays the status of the connection quiesce protocol for the connection. The values are:

Notsupported
The connection quiesce protocol is not supported for this connection.

Unattempted
CICS has not attempted to use the connection quiesce protocol (because the connection has not been released).

Complete
The connection quiesce protocol completed succesfully when the connection was release. The status reverts to UNATTEMPTED when the connection is re-acquired.

Failed
The connection quiesce protocol failed (for instance, becasue a session failed while the protocol was executing). The status reverts to UNATTEMPTED when the connection is re-acquired.

Grname(*value) (appc only***)**
displays (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 system. Otherwise this field is blank. CICS assumes that the partner is a generic resource if the two NETNAMEs sent with a BIND are different. Thus, information may be displayed for a partner that is *not* a generic resource but which uses XRF.

Membername(*value***)**
displays (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 this field is blank. CICS assumes that the partner is a generic resource if the two NETNAMEs sent with a BIND are different. Thus, information may be displayed for a partner that is *not* a generic resource but which uses XRF.

Netname (*value***)**
displays the 8-character name by which the remote system is known to the network.

Note: If the netname is a VTAM LU alias, it is different from the netname component of the NQNAME, which always contains the real netname.

Nqname(*value***)**
displays the 17-character network-qualified name by which the remote system is known to the network.

Pendstatus (*appc and mro only***)**
displays whether there are any pending units of work for this connection. The values are:

Notpending
There has been no mismatch of lognames with the partner.

Note: MRO connections to pre-displays systems do not use lognames. Therefore, for these connections, PENDSTATUS is always 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.

Pendstatus can be reset with Notpending.

For further information about pending units of work, see the *CICS Intercommunication Guide*.

Protocol
displays the protocol in use if this is a VTAM or external CICS interface connection. The values are:

Appc The connection uses the VTAM LUTYPE6.2 protocol for intersystem communication, or is remote.

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 for CICS-to-CICS or CICS-to-IMS intersystem communication.

Purgetype (*input only field***)**
specifies how associated transactions are to be purged. The values are:

Purge (*vtam only and irc only***)**
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.

Forcepurge (*vtam only***)**
All transactions running on sessions on the connected system are immediately terminated abnormally. This can lead to unpredictable results and should be used only in exceptional circumstances.

In some extreme cases (for example, if an error occurs during backout processing), CICS might terminate abnormally.

Cancel
Automatic initiate descriptors (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 will not be canceled. In addition, the following CICS system AIDs are not purged unless FCANCEL is specified.

Table 2. System AIDs requiring FCANCEL to remove them

Description	Tranid
Remote delete AIDs	
Remote scheduler AIDs	CRSR
LU6.2 service manager 1 AIDs	CLS1
LU6.2 service manager 3 AIDs	CLS3
Remote scheduler PURGE AIDs	CRSQ
Resource manager resync AIDs	CRSY

Table 2. System AIDs requiring FCANCEL to remove them (continued)

Description	Tranid
Autoinstalled terminal delete AIDs	CATD
Autoinstalled terminal restart AIDs	CATR

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, the remote AID is canceled asynchronously.

Message DFHTF0100 is written to CSMT to indicate how many AIDs have been deleted for the connection and how many remain.

An 'AIDS CANCELED' message appears on the CEMT panel whenever AIDs are deleted using the CANCEL option of the CEMT SET CONNECTION command.

Fcancel

All AIDs, including system AIDs, queuing for the specified connection are to be canceled. See Table 2 on page 160 for a list of those system AIDS that require FCANCEL to remove them. This can lead to unpredictable results and should be used only in exceptional circumstances.

Note: FCANCEL does not remove transient data AIDs with an associated triggered task. These aids may be removed by purging the associated task.

An 'AIDS CANCELED' message appears on the CEMT panel whenever AIDs are deleted using the FCANCEL option of the CEMT SET CONNECTION command.

Recovstatus (*appc and mro only***)**

displays whether there is resynchronization work outstanding for this connection. The connection may never have been connected, or may have been quiesced and all resynchronization work completed, or disrupted without quiesce—in which case resynchronization may be necessary. The values are:

Norecovdata

Neither side has recovery information outstanding.

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

Recovstatus can be reset with Norecovdata.

Remotesystem (*value***)**

displays the 4-character name of a connection, if the subject of the inquiry is a remote or an indirect connection. In either case, 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.

If the subject of the inquiry is an indirect connection, the value of REMOTESYSTEM is taken from the INDSYS option of the CONNECTION definition.

Otherwise this field is blank.

Rname (*value***)**
displays the 4-character name by which this connection is known in a remote system, if the subject of the inquiry is a remote connection.

Rnetname (*value***)**
displays 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.

Servstatus
displays whether the system can receive and send data. The values are:

Inservice
The system is in service; that is, it is available for use.

Outservice
The system is out of service; that is, it is not available for use.

Note: You can reset this value by overtyping it with a different value.

Uowaction (*input-only field***)**
specifies that the normal resynchronization process is to be partially overridden. The value specifies the type of decision to be taken for any units of work that become in-doubt due to a failure of the connection; the decisions are recorded and any data inconsistencies are reported when the connection is next acquired. The values are:

Backout
All UOWs shunted due to the failure of this connection are to be backed out.

Commit
All UOWs shunted due to the failure of this connection are to be committed.

Forceuow
All UOWs shunted due to the failure of this connection are to be forced to BACKOUT or COMMIT, as specified on the INDOUBT option of the TRANSACTION definition.

Resync
Any UOWs shunted due to 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.

Xlnstatus
displays the status of the exchange lognames (XLN) process. The values are:

Xok (*appc only***)**
The exchange log names (XLN) process for the APPC connection has completed successfully.

Xnotdone (APPC only)
The exchange log names (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.

No status is displayed if exchange log names is not applicable. This can be because the link:

- Is released
- Is MRO, LU6.1, 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*.

CEMT INQUIRE CORBASERVER

Retrieves information about CorbaServers.

Description

INQUIRE CORBASERVER returns information about a particular CorbaServer.

Input

Press the Clear key to clear the screen, and type `CEMT INQUIRE CORBASERVER` (the minimum abbreviation is `CEMT I COR`). You get a display that lists the names and status of the CorbaServer. Note that you cannot change any of the displayed information.

Sample screen

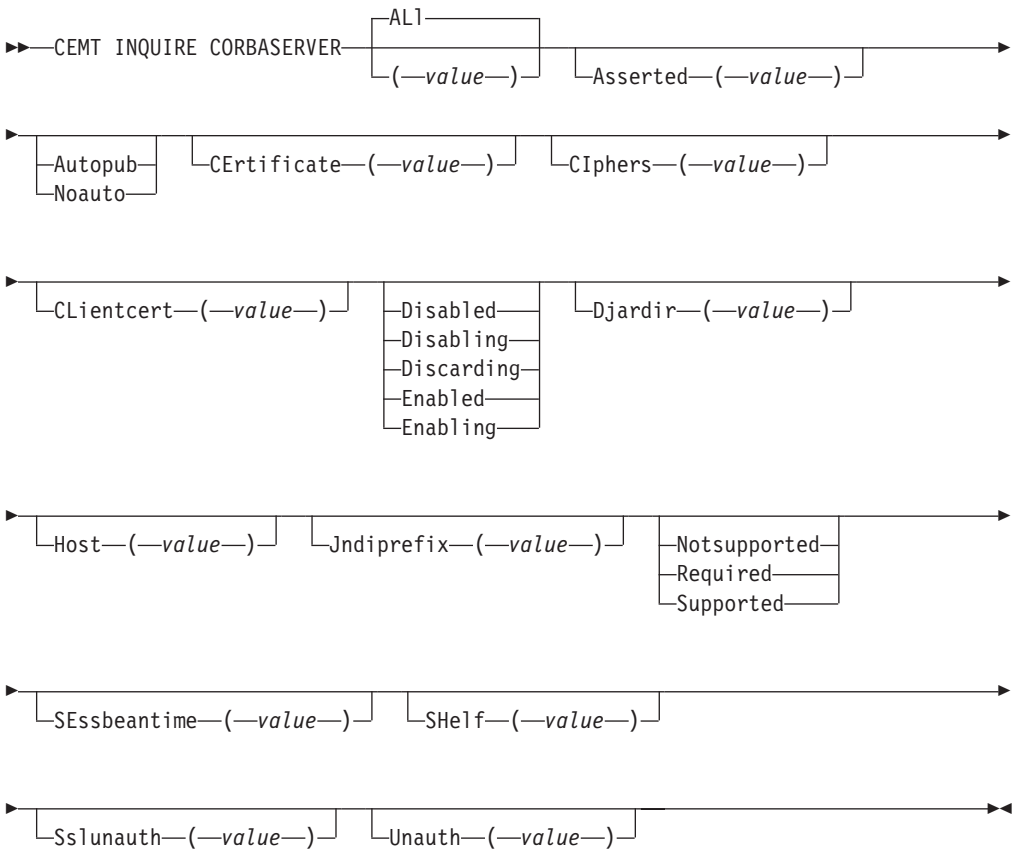
```
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Corba(CBS1) Suppo       Enab  Sessb( 000060 )
    Unaut(IIOPUNA )                Sslun(IIOPSSLU)









                                        SYSID=C2T1 APPLID=IYK4Z2T1
  RESPONSE: NORMAL                     TIME:  11.19.46  DATE: 08.14.01
PF 1 HELP       3 END       5 VAR       7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 73. CEMT INQUIRE CORBASERVER screen

```
i corba
RESULT - OVERTYPE TO MODIFY
  Corbaserver(CBS1)
  Outprivacy(Supported)
  Autopublish( Noauto )
  Enablestatus(Enabled)
  Sessbeantime( 000060 )
  Unauth(IIOPUNA)
  Clientcert()
  Sslunauth(IIOPSSLU)
  Asserted()
  Certificate()
  Jndiprefix()
  Host(winmvs26.hursley.ibm.com)
  Shelf(/var/cicsts/)
  Djardir()
  Ciphers(0504352F0A0903060201)




                                          SYSID=C2T1 APPLID=IYK4Z2T1
                                       TIME:  11.21.52  DATE: 08.14.01
PF 1 HELP 2 HEX 3 END       5 VAR       7 SBH 8 SFH       10 SB 11 SF
```

Figure 74. CEMT INQUIRE CORBASERVER expanded display

INQUIRE CORBASERVER



Conditions: NOTAUTH, NOTFND

(*value***)**
specifies the 1–4 character name of the CorbaServer. The acceptable characters are A-Z a-z 0-9. Do not use names beginning with DFH, because these characters are reserved for use by CICS.

ALl
is the default.

Displayed fields

ASserted(*value***)**
returns the 8-character name of a TCPIPSERVICE that defines the characteristics of the port which is used for inbound IIOP with asserted identity authentication.

AUtopublish(*value***)**
displays a 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 values are:

Autopub
Enterprise beans are to be automatically published.

Noauto
Enterprise beans are not to be automatically published.

CEertificate(*value***)**
specifies the 1–56 character label of the certificate in the key ring that is to be used (as a client certificate) in the SSL handshake for outbound IIOP connections.

If this option is not specified, the default certificate for the key ring is used.

CIphers(*value***)**
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.

CLientcert(*value***)**
returns the 8-character name of a TCPIPSERVICE that defines the characteristics of the port which is used for inbound IIOP with SSL client certificate authentication.

Djardir(*value***)**
displays the 255-character name of the deployed JAR file directory (also known as the *pickup directory*) on z/OS UNIX. (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
displays a value indicating the current state of the CorbaServer. The 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 have 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.

Host(*value***)**
displays the TCP/IP host name, or a string containing the dotted-decimal TCP/IP address, of this logical EJB/CORBA server.

Jndiprefix(*value***)**
specifies a prefix of up to 255 characters to be used at runtime when publishing enterprise beans to the Java Naming and Directory Interface (JNDI). The acceptable characters are A-Z a-z 0-9 . _ /.

The prefix must include any trailing delimiter, such as a forward slash, because CICS does not insert a delimiter between the prefix and a suffix.

If this option is not specified, no prefix is prepended when publishing beans to JNDI.

Outprivacy
displays the level of SSL encryption used for outbound requests from this CorbaServer. The 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(*value***)**
displays, in minutes, the period of inactivity after which a session bean may be discarded by CICS.

0 Session beans will not be timed out.

value Session beans may be discarded after the specified period of inactivity. The maximum value is 357839 (99 days, 23 hours, 59 minutes).

SHelf(*value***)**
specifies the 1–255 character fully-qualified name of a directory (a *shelf*, primarily for *deployed JAR files*) on z/OS UNIX. The acceptable characters are A-Z a-z 0-9 . _ /.

CICS regions into which the CORBASERVER definition is installed must have full permissions to the shelf directory—read, write, and the ability to create subdirectories.

A single shelf directory may be shared by multiple CICS regions and by multiple CORBASERVER definitions. Each CICS region uses a separate subdirectory to

keep its files separate from those of other CICS regions. The subdirectories for CORBASERVER definitions are contained within the subdirectories of the CICS regions into which they are installed. When a CICS region performs a cold or initial start, it deletes its subdirectories from the shelf.

You should not modify the contents of a shelf that is referred to by an installed CORBASERVER definition. If you do, the effects are unpredictable.

Sslunauth
returns the 8-character name of a TCPIPSERVICE that defines the characteristics of the port which is used for inbound IIOP with SSL and no authentication.

Unauth(*value***)**
returns the 8-character name of a TCPIPSERVICE that defines the characteristics of the port which is used for inbound IIOP with no authentication.

CEMT INQUIRE DB2CONN

Function

Retrieves information about the CICS DB2 connection.

Description

INQUIRE DB2CONN returns information about the attributes of the currently installed DB2CONN, which defines the connection to DB2.

Because only one DB2CONN resource can be installed at a time, the name of the DB2CONN is not required on input.

Input

Press the Clear key to clear the screen, and type `CEMT INQUIRE DB2CONN`. The resulting display lists the current status. You can then tab to the highlighted or blank fields and overtype them with the required values.

Sample screen

```
inquire db2conn
STATUS:  RESULTS - OVERTYPE TO MODIFY
  Accountrec( Txid )              Planexitname( DSNCUEXT )
  Authid(         )               Priority( High )
  Authtype( Userid )              Purgecyclem( 00 )
  Comauthid(          )           Purgecycles( 30 )
  Comauthtype( Cuserid )          Resyncmember( Resync )
  Comthreadlim( 0003 )            Signid( CICSHTC1 )
  Comthreads(0000)                Security(              )
  Connecterror( Sqlcode )         Standbymode( Reconnect )
  Connectst( Connected )          Statsqueue( CSSL )
  Db2groupid( DFP2 )              Tcblimit( 0012 )
  Db2id( DF2C )                   Tcbs(0000)
  Db2release(0710)                Threaderror(N906d)
  Drollback(Rollback)             Threadlimit( 0003 )
  Msgqueue1( CSMT )               Threads(0000)
  Msgqueue2(      )               Threadwait( Twait )
  Msgqueue3(      )
  Nontermrel( Norelease )
  Plan(          )
```

Figure 75. CEMT INQUIRE DB2CONN screen

```
►►—CEMT Inquire DB2Conn———————————————————————————————————►◄
```

Displayed fields

Accountrec
Returns the minimum amount of DB2 accounting required for transactions using pool threads. The specified minimum can be exceeded, as described in the following options. The values are as follows:

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.

Task The CICS DB2 attachment facility causes a minimum of one accounting record to be produced for each CICS task.

A transaction that contains multiple UOWs might use a different thread for each of its UOWs (assuming that the thread is released at sync point). The result might be an accounting record produced for each UOW.

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 that contains multiple UOWs uses a different thread for each UOW (assuming the thread is released at sync point. In this case, an accounting record might be produced for each UOW.

None No accounting records are required for transactions using pool threads.

DB2 does, however, produce 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 use for security checking when using pool threads. If an AUTHId is returned, AUTHType is not applicable.

Authtype
Returns the type of ID to use for security checking when using pool threads. If an AUTHType is returned, AUTHid is blank. The values are as follows:

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.

IDs passed to DB2	How DB2 interprets values
CICS sign-on user ID (USERID)	Represents the primary DB2 authorization ID.
RACF® connected group name	If the RACF list of group options is not active, DB2 uses the connected group name supplied by the CICS attachment facility as the secondary DB2 authorization ID. If the RACF list of group options is active, DB2 ignores the connected group name supplied by the CICS attachment facility, but the value appears in the DB2 list of secondary DB2 authorization IDs.

If no RACF group ID is available for this USERID, an 8-character field of blanks is passed to DB2 as the group ID.

Sign The SIGNID parameter of the DB2CONN is used as the resource authorization ID.

Term The terminal identification (four characters padded to eight) is used as an authorization ID. An authorization ID cannot be obtained in this manner if a terminal is not connected with the transaction.

If a transaction is started (using a CICS command) and has no terminal associated with it, 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 user ID, 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 use for security checking when using command threads. If COMAUTHType is returned, COMAUTHid is not applicable.

Comauthtype

Returns the type of ID to use for security checking when using command threads. If COMAUTHType is returned, COMAUTHid is blank. The values are as follows:

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.

IDs passed to DB2	How DB2 interprets values
CICS sign-on user ID (USERID)	Represents the primary DB2 authorization ID.
RACF connected group name	If the RACF list of group options is not active, DB2 uses the connected group name supplied by the CICS attachment facility as the secondary DB2 authorization ID. If the RACF list of group options is active, DB2 ignores the connected group name supplied by the CICS attachment facility, but the value appears in the DB2 list of secondary DB2 authorization IDs.

If no RACF group ID is available for this USERID, an 8-character field of blanks is passed to DB2 as the group ID.

Csign

The SIGNID parameter of the DB2CONN resource 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 user ID 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).

Comthreadlim
Returns the current maximum number of command threads the CICS DB2 attachment allows active before requests overflow to the pool.

Comthreads
Returns the current number of active command threads.

Connecterror
Returns a value when CICS is not connected to DB2 because the adapter is in standby mode. The value shows how this information is to be passed back to an application that has issued a SQL request. The values are as follows:

Abend
The application is abended with abend AEY9.

Sqlcode
The application receives a -923 SQLCODE.

Connectst
Returns the status of the CICS DB2 connection. The values are as follows:

Connected
CICS is connected to DB2.

Notconnected
CICS is not connected to DB2.

Db2groupid
If you are using group attach, this returns the name (up to four characters) of a data sharing group, or subgroup of DB2 subsystems that you have specified. CICS attempts to connect to any active member of this group. If the CICS DB2 attachment is connected, the name of the DB2 subsystem that was chosen from the group appears in the Db2id field. If CICS is waiting to reconnect to a specific DB2 subsystem within the data sharing group for which it is holding outstanding units of work, the name of the specific DB2 subsystem appears in the Db2id field, and CEMT displays the message “Waiting for DB2”. (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(Resync) is specified and the DB2 subsystem for which CICS is holding outstanding units of work is unavailable, this field gives the name of the DB2 subsystem, and CEMT displays the message “Waiting for DB2”.

Db2release
Returns a four-character value indicating the version and release level of the DB2 susbsytem 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 if a transaction is selected as victim of a deadlock resolution. The values are as follows:

Rollback
the attachment facility issues a sync point rollback before returning control to the application. An SQL return code of -911 is returned to the program.

Norollback
the attachment facility does not 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.

Nontermrel
Returns a value showing whether non-terminal transactions are to release threads for reuse at intermediate syncpoints. CVDA values are as follows:

Release
Non-terminal transactions release threads for reuse at intermediate sync points.

Norelease
Non-terminal transactions do not release threads for reuse at intermediate sync points.

Plan
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). The thread TCBs are CICS open L8 TCBs. CVDA values are as follows:

HIGH Thread TCBs have a higher priority than the CICS QR TCB.

EQUAL
Thread TCBs have equal priority with the CICS QR TCB.

LOW Thread TCBs have a lower priority than the CICS QR TCB.

Purgecyclem
Returns the number of minutes in the protected thread purge cycle time (the number of seconds is returned by PURGECYCLES). The range for PURGECYCLEM is 0 - 59.

A protected thread is not terminated immediately when it is released. It is terminated only after two completed purge cycles, if it has not been reused in the meantime. For example, if the protected thread purge cycle is set to 30 seconds, a protected thread is purged 30 - 60 seconds after that thread is released. An unprotected thread is terminated when it is released (at sync point or end of task) if there are no other transactions waiting for a thread on that DB2ENTRY resource.

Purgecycles
Returns the number of seconds in the protected thread purge cycle time (the number of minutes is returned by PURGECYCLEM). The range for PURGECYCLES is 0 - 59. If PURGECYCLEM is zero, the minimum value of PURGECYCLES is 30 seconds.

A protected thread is not terminated immediately when it is released. It is terminated only after two completed purge cycles, if it has not been reused in the meantime. For example, if the protected thread purge cycle is set to 30 seconds, a protected thread is purged 30 - 60 seconds after that thread is released. An unprotected thread is terminated when it is released (at sync point or end of task) if there are no other transactions waiting for a thread on that DB2ENTRY resource.

Resyncmember
This applies only if you are using group attach (with a DB2 groupid specified), and Returns the strategy that CICS adopts if outstanding units of work are being held for the last DB2 data sharing group member to which CICS was connected. (Units of work that are shunted indoubt are not included in this process, because CICS itself is unable to resolve those units of work at this time. Resynchronization for those UOWs occurs when CICS has resynchronized with its remote coordinator.) If you are not using group attach, a blank field is returned. CVDA values are as follows:

Resync
CICS connects to the same DB2 data sharing group member.

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.

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(SIGNID) and command threads specifying COMAUTHTYPE(CSIGN).

Security
This field has no meaning for INQUIRE. It is supplied to allow overtyping with the REBUILD option, to initiate a CEMT SET DB2CONN command. See SECURITY.

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. The values are as follows:

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 then fails, the CICS DB2 attachment reverts again to standby mode and later reconnects to DB2 when DB2 becomes available.

Statsqueue
Returns the transient data destination for CICS DB2 attachment statistics produced when the CICS DB2 attachment is shut down.

Tcbs
Returns a number indicating the TCBs currently used by the CICS DB2 attachment facility. 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. CICS creates open TCBs (up to the limit specified in the system initialization parameter MAXOPENTCBS). The TCBLIMIT attribute of the DB2CONN definition governs how many of the open TCBs can be used to access DB2; that is, how many of them can identify to DB2 and create a connection into DB2.

Threaderror
Returns the processing that is to occur following a create thread error. The values are as follows:

Abend
For a second or subsequent SQL error, the transaction abends 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 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 a value showing whether transactions should wait for a pool thread or be abended if the number of active pool threads reaches the threadlimit number. The 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.

CEMT 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.

Description

The entry is identified by the name with which it was defined in CEDA.

Input

For RCTs migrated to the CSD, the name is the name of the first transaction on the DSNCRCT TYPE=ENTRY statement unless the RDONAME parameter has been specified.

Sample screen

```
inquire db2entry
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Db2e(DB2ENT1 )     Use Ena Poo Hig Pro( 0000 ) Pth(0000)
    Threadl( 0000 ) Threads(0000) Tpo                 Plane( DSNCUEXT
 Db2e(DB2ENT2 )     Use Ena Poo Hig Pro( 0000 ) Pth(0000)
    Threadl( 0000 ) Threads(0000) Tpo                 Plane( DSNCUEXT
```

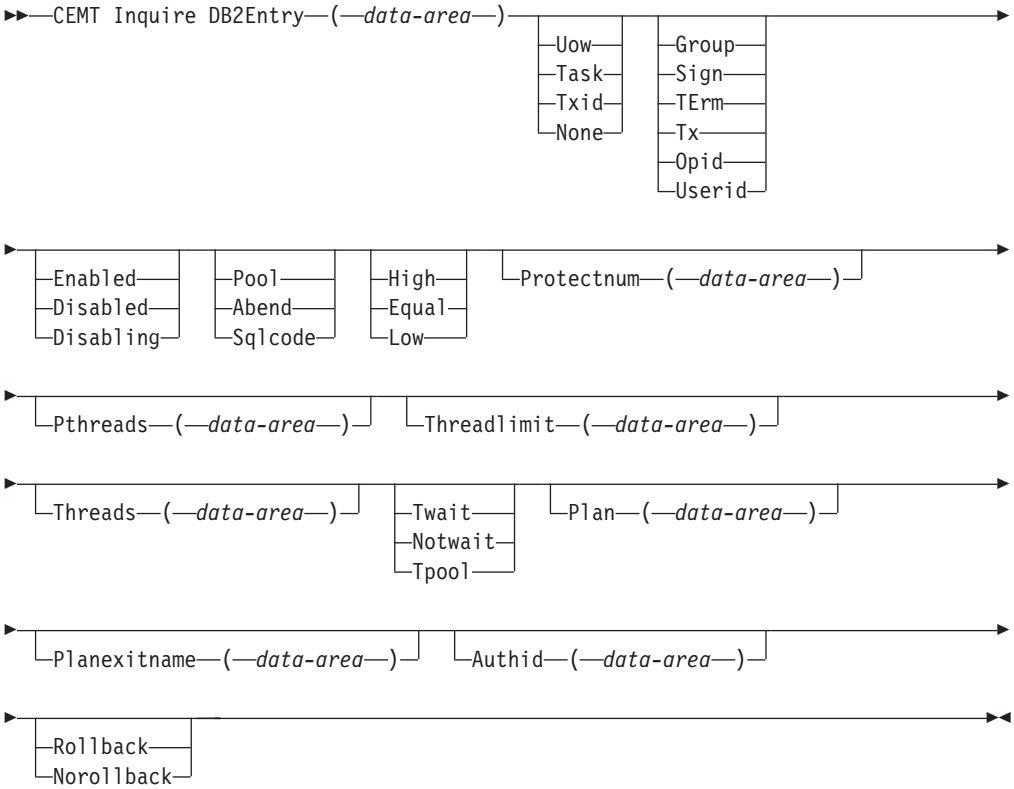
Figure 76. CEMT INQUIRE DB2ENTRY screen

If you place the cursor against the result and press ENTER, an expanded format is displayed as shown in Figure 77 on page 175.

```
inquire db2entry
RESULT - OVERTYPE TO MODIFY
  Db2entry(DB2ENT1)
  Accountrec( None )
  Authtype( Userid )
  Enablestatus( Enabled )
  Disabledact( Pool )
  Priority( High )
  Protectnum( 0000 )
  Pthreads(0000)
  Threadlimit( 0000 )
  Threads(0000)
  Threadwait( Tpool )
  Plan(          )
  Planexitname( DSNCUEXT )
  Authid(          )
  Drollback(Rollback)
```

Figure 77. The expanded display of an individual entry

CEMT INQUIRE DB2ENTRY



Displayed fields

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

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.

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 may use a different thread for each of its UOWs (assuming the thread is released at syncpoint). The result may be an accounting record produced for each UOW.

Txid The CICS DB2 attachment facility causes an accounting record to be produced 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.

None No accounting records are required for transactions using threads from this DB2ENTRY.

DB2 does, however, produce 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 for threads on this DB2ENTRY. If an AUTHType is returned, AUTHid is blank. The 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.

IDs passed to DB2	How DB2 interprets values
CICS sign-on user ID (USERID)	Represents the primary DB2 authorization ID.
RACF connected group name	If the RACF list of group options is not active, then DB2 uses the connected group name supplied by the CICS attachment facility as the secondary DB2 authorization ID. If the RACF list of group options is active, DB2 ignores the connected group name supplied by the CICS attachment facility, but the value appears in the DB2 list of secondary DB2 authorization IDs.

If no RACF group ID is available for this 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.

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 a value showing what CICS is to do with new transactions accessing DB2ENTRY when it has been disabled or is disabling. If DISABLEDACT is not specified, and DB2ENTRY is disabled, new requests are routed to the pool by default. The 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 returns a value showing whether the CICS DB2 attachment should initiate a SYNCPOINT rollback if a transaction is selected as victim of a deadlock resolution. The 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 does not 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. The 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 cannot access the DB2ENTRY. Existing transactions using the DB2ENTRY will be allowed to complete unless the DB2ENTRY is being disabled with the FORCE option.

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 pool thread TCBs relative to the CICS main TCB (QR TCB). The thread TCBs are CICS open L8 TCBs. CVDA values are:
HIGH Thread TCBs have a higher priority than the CICS QR TCB.
EQUAL
Thread TCBs have equal priority with the CICS QR TCB.
LOW Thread TCBs have a lower priority than the CICS QR TCB.

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.

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).

Threads
returns the current number of threads active for this DB2ENTRY.

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. The 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.

CEMT INQUIRE DB2TRAN

Returns attributes of a particular DB2TRAN that 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.

Input

If a TRANSID is specified on a DB2ENTRY when the DB2ENTRY is installed, CICS installs a DB2TRAN named DFHtttt, where tttt is the TRANSID.

Sample screen

```
inquire db2tran
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Db2t(DFHTXN* ) Db2e( DB2ENT2  ) Tra( TXN* ) Plane(DSNCUEXT)
 Db2t(DFHTXN2 ) Db2e( DB2ENT1  ) Tra( TXN2 ) Plane(DSNCUEXT)
 Db2t(TXN1    ) Db2e( DB2ENT1  ) Tra( TXN1 ) Plane(DSNCUEXT)
```

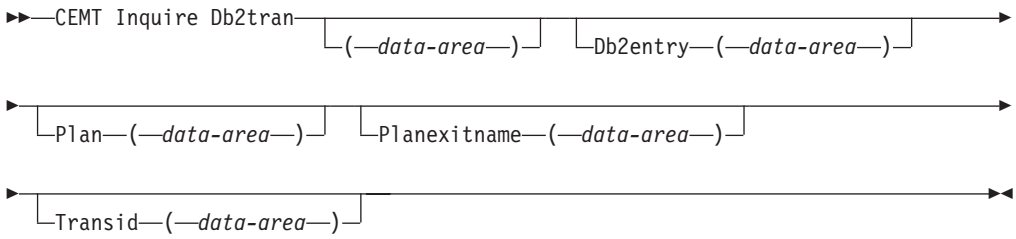
Figure 78. CEMT INQUIRE DB2TRAN screen

If you place the cursor against the result and press ENTER, an expanded format is displayed as shown in Figure 79.

```
inquire db2tran
RESULT - OVERTYPE TO MODIFY
  Db2tran(DFHTXN*)
  Db2entry( DB2ENT2  )
  Transid( TXN* )
  Plan()
  Planexitname(DSNCUEXT)
```

Figure 79. The expanded display of an individual entry

CEMT INQUIRE DB2TRAN



Displayed fields

Db2entry
returns the name of the DB2ENTRY to which this DB2TRAN refers. This is the DB2ENTRY with which this additional transaction should be associated.

Plan
returns the name of the plan retrieved from the associated DB2ENTRY if it exists. If there is no 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*.

CEMT INQUIRE DELETSHIPPED

Display system settings that control the CICS timeout delete mechanism.

Description

The CICS timeout delete mechanism is invoked at user-specified intervals to remove any shipped terminal definitions that have not been used for longer than a user-specified time.

INQUIRE DELETSHIPPED displays the current settings of the parameters that control the mechanism.

Input

Press the Clear key and type `CEMT INQUIRE DELETSHIPPED` (the minimum abbreviation is `CEMT I DE`). You will get a display screen.

To change attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field.
- Use the CEMT SET DELETSHIPPED command.

Sample screen

```
IN DELE
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Del Int( 120000 ) Idl( 020000 )
```

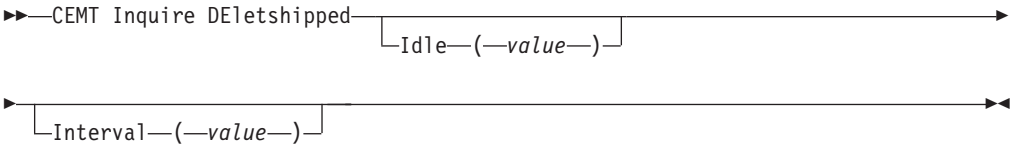
Figure 80. CEMT INQUIRE DELETSHIPPED screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 81.

```
IN DELE
RESULT - OVERTYPE TO MODIFY
  Deletshipped
  Interval( 120000 )
  Idle( 020000 )
```

Figure 81. The expanded display

CEMT INQUIRE DELETSHIPPED



Displayed fields

Del
identifies this panel as relating to CICS timeout delete information.

Idle(*value*)
displays, in the form "**0hhmmss+**", the minimum time that an *inactive* shipped terminal definition must remain installed in this region. When the CICS timeout delete mechanism is invoked, only those shipped definitions that have been inactive for longer than this time are deleted.

The time interval can be in the range 00-99 hours; 00-59 minutes; and 00-59 seconds.

Note: You can reset this value by overtyping it with a different value.

Interval(*value*)
displays, in the form "**0hhmmss+**", the interval between invocations of the CICS timeout delete mechanism. The timeout delete mechanism removes any shipped terminal definitions that have not been used for longer than the time displayed by the IDLE option.

The time interval can be in the range 00-99 hours; 00-59 minutes; and 00-59 seconds.

Note: You can reset this value by overtyping it with a different value.

CEMT INQUIRE DISPATCHER

Retrieve system information used by the CICS dispatcher.

Syntax

Press the Clear key and type `CEMT INQUIRE DISPATCHER` (the minimum abbreviation is `CEMT I DI`). You will get a display screen.

To change attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field.
- Use the CEMT SET DISPATCHER command.

Sample screen

```
 inq dispatcher
  STATUS:  RESULTS - OVERTYPE TO MODIFY
Actjvmtcbs(000)
Actopentcbs(000)
Actssltcbs(000)
Actxptcbs(000)
Aging( 32768 )
Maxjvmtcbs( 005 )
Maxopentcbs( 005 )
Maxssltcbs(008)
Maxxptcbs(005)
Mrobatch( 001 )
Runaway( 0005000 )
Scandelay( 0500 )
Subtasks(000)
Time( 0001000 )
                                    SYSID=HT61 APPLID=CICSHT61
   RESPONSE: NORMAL                TIME:  12.39.18  DATE: 07.30.01
 PF 1 HELP       3 END       5 VAR       7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 82. CEMT INQUIRE DISPATCHER screen

CEMT INQUIRE DISPATCHER

Options

ACTJvmtcbs(*value***)**
displays the total number of J8 and J9 mode open TCBs that are currently active (that is, allocated to a user task).

ACTOpentcbs(*value***)**
displays the number of L8 and L9 mode open TCBs that are currently active (that is, allocated to a user task).

ACTSsltcbs(*value***)**
displays the number of S8 mode open TCBS that are currently active (that is, allocated to a user task).

ACTXptcbs(*value***)**
displays the number of X8 and X9 mode open TCBs that are currently active (that is, allocated to a user task).

AGing(*value***)**
displays the time factor for CICS to use in the priority aging algorithm for incrementing the priority of a task. The value is expressed as “milliseconds per unit of priority”, and is in the range 0–65535.

The priority aging factor is defined initially by the PRTYAGE system initialization parameter, in which the value represents the number of milliseconds that must elapse before the priority of a waiting task can be adjusted upward by 1. See the *CICS System Definition Guide* for information about the **PRTYAGE** parameter.

Note: You can reset this value by overtyping it with a different value.

MAXJvmtcbs(*value***)**
displays the maximum number of J8 and J9 mode open TCBs that CICS is allowed to attach. The value can be in the range 1–999.

Note: You can reset this value by overtyping it with a different value.

MAXOpentcbs(*value***)**
displays the maximum number of L8 and L9 mode open TCBs that can exist concurrently in the CICS region. The value can be in the range 1–2000.

Note: You can reset this value by overtyping it with a different value.

MAXSsltcbs(*value***)**
displays the maximum number of S8 mode open TCBs that can exist concurrently in the CICS region. The value can be in the range 1–1024.

Note: You can reset this value by overtyping it with a different value.

MAXXptcbs(*value***)**
displays the maximum number of X8 and X9 mode open TCBs that can exist concurrently in the CICS region. The value can be in the range 1–999.

Note: You can reset this value by overtyping it with a different value.

MRobatch(*value*)
displays the number of MRO requests from connected regions that are to be batched before this region is posted. It is a value in the range 1–255.

Note: You can reset this value by overtyping it with a different value.

Runaway(*value*)
displays the interval, in milliseconds, for which a task can have control before it is assumed to be looping.

Note: You can reset this value by overtyping it with a different value.

If you are resetting this value, you can specify 0, or a value in the range 500–2700000. The value you specify is rounded down to a multiple of 500.

CICS purges a task if the task has not given up control after this interval (that is, CICS assumes that the task is looping). If you specify zero, runaway task control is inoperative (that is, tasks are not purged if they appear to be looping). The value is independent of, and can be less than, the value in the TIME operand.

CICS runaway-task detection is based upon task time (that is, the interval is reset each time a task receives control of the processor, and is then decremented while the task is in control). You do not, therefore, need to allow for the effect of other jobs when setting the runaway value.

SCandelay(*value*)
displays the maximum number of milliseconds between the receipt of a VTAM request, and the time when CICS begins to process the request. The number is in the range 0–5000, and is less than or equal to the TIME value.

Note:

1. You can reset this value by overtyping it with a different value.
2. In the summary report of the dispatcher statistics, SCANDELAY is referred to as ICVTSD.

SUbtasks(*value*)
displays the value set by the SUBTSKS system initialization parameter, which can be either 0 or 1.

Time(*value*)
displays the interval, in milliseconds, for which CICS releases control to the operating system if no transactions are ready to resume processing. This interval is known as the ‘region exit interval’.

Note:

1. You can reset this value by overtyping it with a different value. It can have a value in the range 100–3600000, and must be greater than or equal to the SCANDELAY value.
2. In the summary report of the dispatcher statistics, TIME is referred to as ICV.

CEMT INQUIRE DJAR

Retrieves information about a DJAR definition.

Description

INQUIRE DJAR returns information about any currently installed DJAR definition.

Input

Press the Clear key to clear the screen, and type CEMT INQUIRE DJAR (the minimum abbreviation is CEMT I DJ). You get a display that lists the names and status of any deployed JAR archive files. Note that you cannot change any of the displayed information.

Sample screen

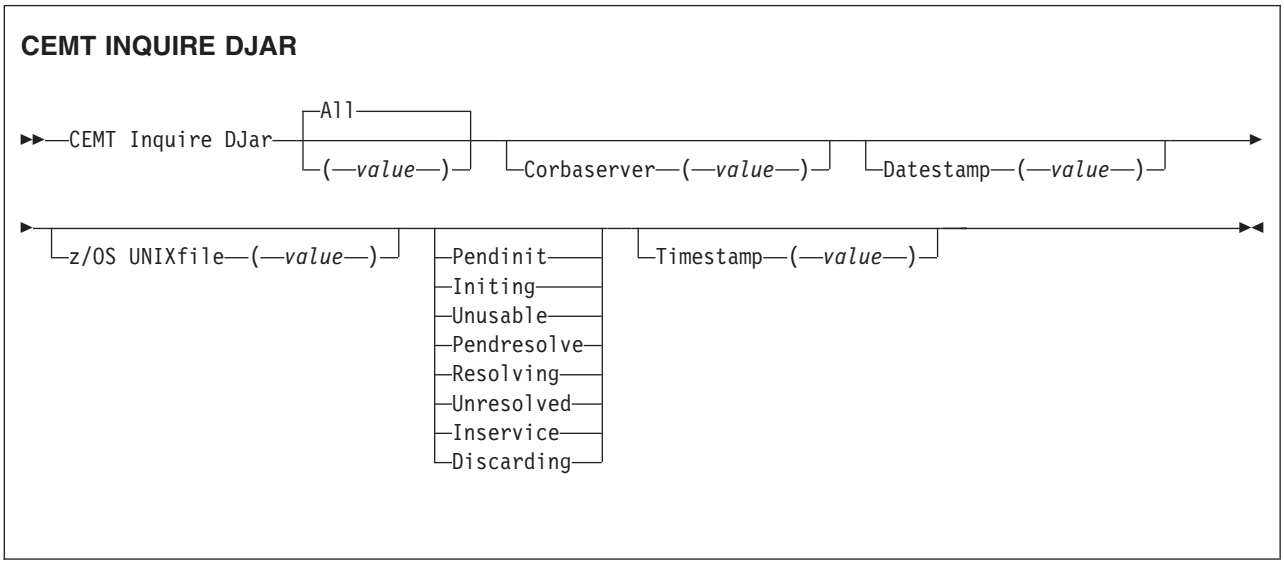
```
inq djar
STATUS:  RESULTS
 Djar(BJAR                            ) Corba(CORB) Inser
    Dates(20010221) Times(11:29:41) z/OS UNIXfi(/u/ivanh/ivanejbd.jar     )
 Djar(IJAR                            ) Corba(CORA) Inser
    Dates(20010222) Times(14:33:52) z/OS UNIXfi(/u/ivanh/ivanejbd.jar     )


                                          SYSID=HA61 APPLID=CICSHA61
  RESPONSE: NORMAL                       TIME:  15.54.22  DATE: 09.03.98
PF 1 HELP       3 END       5 VAR       7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 83. CEMT INQUIRE DJAR screen

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format.

```
 inq djar
 RESULT
   Djar(BJAR)
   Corbaserver(CORB)
   State(Inservice)
   Datestamp(20010923)
   Timestamp(17:39:48)
   z/OS UNIXfile(/u/ivanh/ivanejbd.jar)
```

Figure 84. The expanded display of an individual entry



(*value***)**
is a 1–32 character deployed JAR file name.

All
is the default.

Displayed fields

Corbaserver(*value***)**
displays a 4-character CorbaServer name.

Datestamp(*value***)**
displays the date, in yyyymmdd format, of the deployed JAR file on z/OS UNIX. This is a readonly value that CICS updates when the DJAR resource is installed or updated. You can use the DATESTAMP and TIMESTAMP values to determine whether CICS has refreshed itself after you've made an update to a JAR in the deployed JAR file (pickup) directory.

For dynamically-installed DJARs (those installed by the CICS scanning mechanism), the value of DATESTAMP is the date of the z/OS UNIX 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 DATESTAMP is the date of the z/OS UNIX file pointed to by the DJAR definition, at the time the DJAR was installed.

If the INQUIRE DJAR command is executed before the DJAR resource has fully initialized, this field displays blanks.

z/OS UNIXfile(*value***)**
displays a 255-character, fully-qualified z/OS UNIX file name.

State(*value***)**
The values are:

Pendinit
The DJar has just been created and is awaiting Stage 1 Resolution.

Initing
The DJar is undergoing Stage 1 Resolution Processing (or has got stuck in this phase), see EJIO which involves copying the DJar to the Shelf.

Unusable
The DJar has failed Stage 1 Resolution and is not available for use.

Pendresolve
The DJar has undergone Stage 1 Resolution correctly (or was recovered on a Warm start after resolving correctly in a prior region run) and is awaiting Stage 2 Resolution processing (Bean loading).

Resolving
The DJar is undergoing Stage 2 Resolution Processing (or has got stuck in this state, see EJJO) and so its Beans are being loaded by CEJR.

Unresolved
The DJar has failed Stage 2 Resolution, or some other processing, and is not available for use; Beans were not correctly loaded).

Inservice
The DJar has undergone both Stage 1 and Stage 2 Resolution and is available for use; its Beans are all correctly loaded as well.

Discarding
The DJar is being deleted.

Timestamp(*value***)**
displays the time, in hh:mm:ss format, of the deployed JAR file on z/OS UNIX.

This is a readonly value that CICS updates when the DJAR resource is installed or updated. You can use the DATESTAMP and TIMESTAMP values to determine whether CICS has refreshed itself after you've made an update to a JAR in the deployed JAR file (pickup) directory.

For dynamically-installed DJARs (those installed by the CICS scanning mechanism), the value of TIMESTAMP is the time of the z/OS UNIX 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 TIMESTAMP is the time of the z/OS UNIX file pointed to by the DJAR definition, at the time the DJAR was installed.

If the INQUIRE DJAR command is executed before the DJAR resource has fully initialized, this field displays blanks.

CEMT INQUIRE DOCTEMPLATE

Retrieves information about a DOCTEMPLATE resource definition for a CICS document template.

Description

INQUIRE DOCTEMPLATE returns information about any currently installed document templates.

Input

Press the Clear key to clear the screen, and type `CEMT INQUIRE DOCTEMPLATE` (the minimum abbreviation is `CEMT I DO`). You get a display that lists the names and status of any document templates. Note that you cannot change any of the displayed information.

Sample screen

```
I DOC
STATUS:  RESULTS - OVERTYPE TO MODIFY
   Doc(BROWSER ) Mem Nam(BROWSER         ) App Ebc
      Tem(BROWSER                   ) Ddn(DFHHTML ) Siz(0000000412)
   Doc(DFHWBPW1) Pro Nam(DFHWBPW1        ) App Ebc
      Tem(DFHWBPW1                  )               Siz(0000000000)
   Doc(SHOWNAME) Hfs Nam(showname.html   ) App Ebc
      Tem(showname                  )               Siz(0000000180)
   Doc(WIRELESS) Hfs Nam(wireless.xml    ) App Ebc
      Tem(WIRELESS                  )               Siz(0000002133)
```

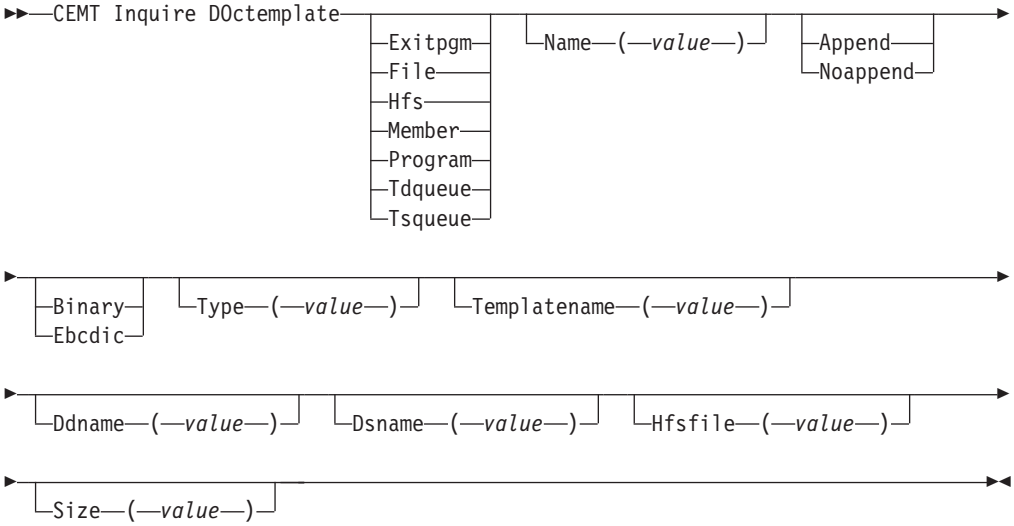
Figure 85. CEMT INQUIRE DOCTEMPLATE screen

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format.

```
i doc
RESULT - OVERTYPE TO MODIFY
  Doctemplate(WIRELESS)
  Templatetype(Hfs)
  Name(wireless.xml)
  Appendcrlf(Append)
  Type(Ebcdic)
  Templatename(WIRELESS)
  Ddname()
  Dsname()
  Hfsfile(/u/coultha/website/wireless.xml)
  Size(0000002133)
```

Figure 86. The expanded display of an individual entry

CEMT INQUIRE DOCTEMPLATE



Displayed fields

Appendcrlf
returns whether CICS is to delete trailing blanks from, and append carriage-return line-feed to, each logical record of the template.

Append
append carriage-return and line-feed.

Noappend
do not append carriage-return and line-feed.

DDname
returns the DD name of the PDS containing the template. The DD name applies only to a template of type PDS. If a member name is supplied without a value for DDNAME, the default value DFHHTML is used.

DOctemplate
returns the name of this DOCTEMPLATE resource definition.

DSname
returns the DS name of the PDS containing the template.

Hfsfile

returns the fully-qualified name of the z/OS UNIX file in which the template resides. The Hfsfile attribute applies only to a template of type HFS. It can be up to 255 characters in length.

Name
returns the name of the location defined in TEMPLATETYPE.

Size
returns the amount of storage, in bytes, used by the cached copy of the document template. A value of zero is returned if there is no cached copy of the template at the time of the enquiry.

Templatename
returns the extended template name by which the document template is known outside the resource definition function (the TEMPLATENAME attribute of the DOCTEMPLATE resource definition).

Templatetype
returns the type of resource whose name is returned in NAME.

Exitpgm
An exit program.

File A CICS file name for a data set.

Hfs A z/OS UNIX file.

Member
A name of the member in the PDS described in DDNAME.

Program
A program.

Tdqueue
A TD queue.

Tsqueue
A TS queue.

Type
returns the format of the template contents.

Binary
When the template is loaded from the template library, no parsing of the template's contents is done.

Ebcdic
When the template is loaded from the template library, the contents are parsed as EBCDIC text.

CEMT INQUIRE DSAS

Retrieve information about the local CICS dynamic storage areas (DSAs).

Description

CEMT INQUIRE DSAS returns the value of various DSA sizes and limits.

Input

Press the Clear key and type `CEMT INQUIRE DSAS` (the minimum abbreviation is `CEMT I DSA`).

To change various attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see Figure 87)
- Use the CEMT SET DSAS command.

Sample screen

```
IN DSAS
STATUS:  RESULTS - OVERTYPE TO MODIFY
   Sosabovebar(Notsos)
   Sosaboveline(Notsos)
   Sosbelowline(Notsos)

  Dsalimit( 04194304 )
   Cdsasize( 00524288 )
   Rdsasize( 00262144 )
   Sdsasize( 00262144 )
   Udsasize( 00000000 )

  Edsalimit( 0020971520 )
   Ecdsasize( 0002097152 )
   Erdsasize( 0003145728 )
   Esdsasize( 0000000000 )
   Eudsasize( 0001048576 )

  Memlimit(Nolimit)
```

Figure 87. CEMT INQUIRE DSAS screen

CEMT INQUIRE DSAS

Displayed fields

Cdsasize(*value***)**
displays the size, in bytes, of the CICS dynamic storage area (CDSA). The size of this storage area is calculated and managed by CICS automatically, within the overall limits specified for all the DSAs that reside below the 16MB boundary.

Dsalimit(*value***)**
displays the maximum amount of storage, as a total number of bytes, within which CICS can dynamically allocate storage for the four individual DSAs that reside below the 16MB boundary. (See the *CICS System Definition Guide* for a description of the DSALIM system initialization parameter.)

Note: You can reset this value by overtyping it with a different value.

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 dynamic storage is freed in the individual DSAs.

Ecdsasize(*value***)**
displays the size, in bytes, of the extended CICS dynamic storage area (ECDSA) above the 16MB boundary. The size of this storage area is calculated and managed by CICS automatically, within the overall limits specified for all the DSAs that reside above the 16MB boundary.

Edsalimit(*value***)**
displays the maximum amount of storage, as a total number of bytes, within

which CICS can dynamically allocate storage for the four individual DSAs that reside above the 16MB boundary. (See the *CICS System Definition Guide* for a description of the EDSALIM system initialization parameter.)

Note: You can reset this value by overtyping it with a different value.

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 dynamic storage is freed in the individual DSAs.

Erdsasize(*value***)**
displays the size, in bytes, of the extended read-only dynamic storage area (ERDSA) above the 16MB boundary. The size of this storage area is calculated and managed by CICS automatically, within the overall limits specified for all the DSAs that reside above the 16MB boundary.

Esdsasize(*value***)**
displays the current size of the extended shared dynamic storage area (ESDSA). The size of this storage area is calculated and managed by CICS automatically, within the overall limits specified for all the DSAs that reside above the 16MB boundary.

Eudsasize(*value***)**
displays the size, in bytes, of the extended user dynamic storage area (EUDSA) above the 16MB boundary. The size of this storage area is calculated and managed by CICS automatically, within the overall limits specified for all the DSAs that reside above the 16MB boundary.

Memlimit(*value***)**
displays the amount of storage available above the 2GB boundary (above the bar), for use by the CICS region. A value of NOLIMIT indicates that no limit has been imposed on the amount of storage that the region can attempt to use.

Rdsasize(*value***)**
displays the current size of the read-only dynamic storage area (RDSA). The size of this storage area is calculated and managed by CICS automatically, within the overall limits specified for all the DSAs that reside below the 16MB boundary.

Sdsasize(*value***)**
displays the current size of the shared dynamic storage area (SDSA). The size of this storage area is calculated and managed by CICS automatically, within the overall limits specified for all the DSAs that reside below the 16MB boundary.

Sosabovebar(*value***)**
displays whether CICS is short-on-storage in the dynamic storage areas above the 2GB boundary (above the bar).
Notsos
CICS is not short-on-storage in any of the dynamic storage areas above the 2GB boundary.
Sos CICS is short-on-storage in at least one of the dynamic storage areas above the 2GB boundary.

Sosaboveline(*value***)**
displays whether CICS is short-on-storage in the dynamic storage areas above the 16MB line, but below the 2GB boundary.
Notsos
CICS is not short-on-storage in any of the dynamic storage areas above the 16MB line (but below the 2GB boundary).

Sos CICS is short-on-storage in at least one of the dynamic storage areas above the 16MB line (but below the 2GB boundary).

Sosbelowline(*value***)**
displays whether CICS is short-on-storage in the dynamic storage areas below the 16MB line.

Notsos
CICS is not short-on-storage in any of the dynamic storage areas below the 16MB line.

Sos CICS is short-on-storage in at least one of the dynamic storage areas below the 16MB line.

Udsasize(*value***)**
displays the size, in bytes, of the user dynamic storage area (UDSA) below the 16MB boundary.

CEMT INQUIRE DSNAME

Retrieve information about an external data set.

Description

INQUIRE DSNAME returns information about the status of a data set managed by CICS file control. The data set name is either the name specified on the DSN parameter of the data set's DD statement, or the DSNAME parameter of the data set's file resource definition. Its main purpose is to allow you to monitor failing VSAM data sets. Many of the attributes cannot be determined until a file that references the data set has been opened. If this has not happened, blank values are returned for most attributes.

Note: CEMT has to retrieve QUIESCESTATE information from the ICF catalog. This can slow down processing of this command. QUIESCESTATE is not supplied by default but is retrieved and displayed if requested.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE DSNAME` (the minimum abbreviation is `CEMT I DSN`). You get a display that lists the current status except QUIESCESTATE. To retrieve QUIESCESTATE you can place the cursor at the beginning of the displayed item (in front of the Dsn field) and press the enter key.
- Type `CEMT INQUIRE DSNAME` (`CEMT I DSN`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i dsn vs not`, the resulting display will show you the details of only those external data sets that are VSAM data sets and are not recoverable.

You can then tab to the highlighted or blank fields and overtype them with the required values.

ALl
is the default.

(*value***)**
specifies one or more names (1-44 characters) of an external data set.

Sample screen

```
IN DSN
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Dsn(CICSTS32.CICS.CICSH###.DFHCSD                    ) Vsa
    Fil(0001) Val Bas     Sta         Ava
 Dsn(CICSTS32.CICS.CICSHT61.FILEA                     ) Vsa
    Fil(0001) Val Bas     Sta         Ava
```

Figure 88. CEMT INQUIRE DSNAME screen

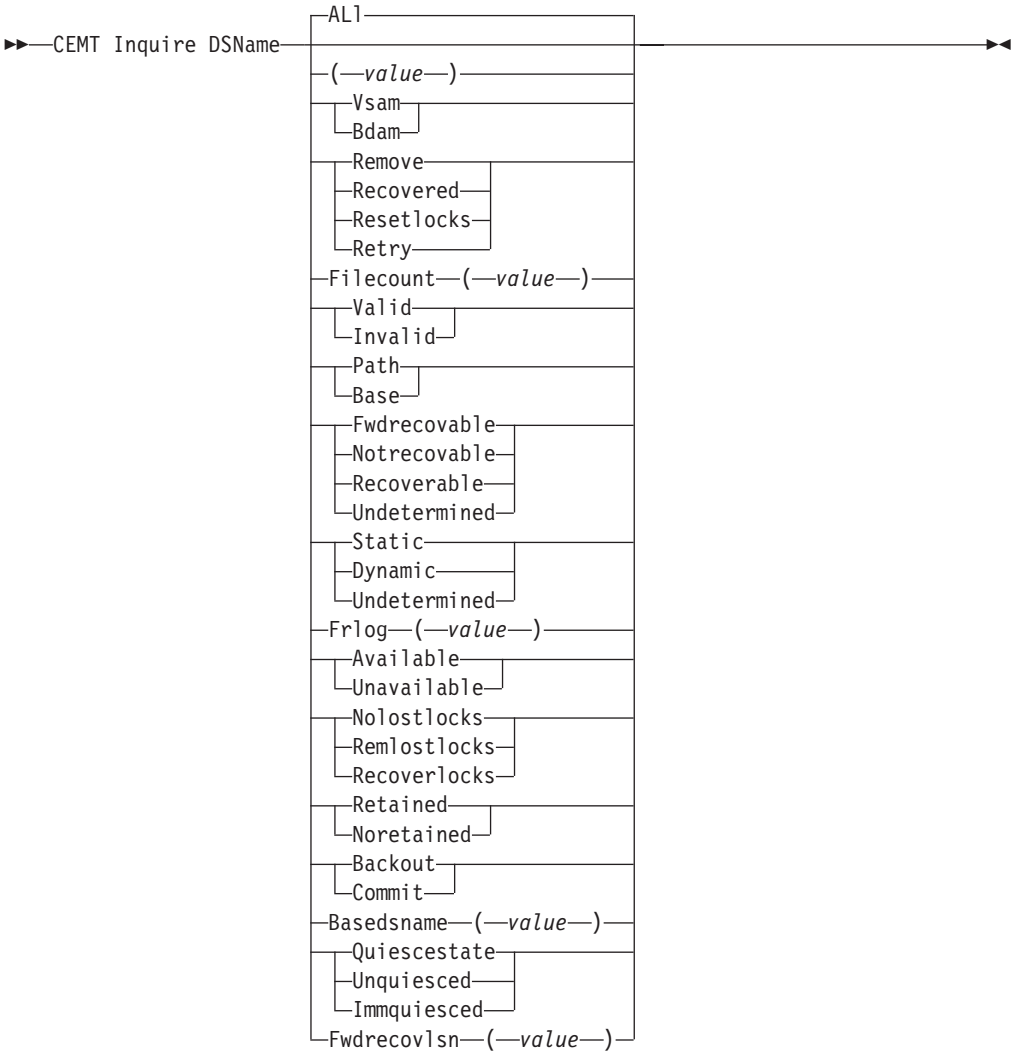
Note: There are blank fields on the screen. With the exception of QUIESCESTATE, these show where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

Because of the large number of keywords supported for this command, only two lines of information for each result are displayed. The full set of keywords can be displayed by moving the cursor to the beginning of a displayed item (in front of the Dsn field) and pressing the ENTER key. You get a display of all the keyword information for the displayed item, as follows:

```
IN DSN
RESULT - OVERTYPE TO MODIFY
  Dsname(CICSTS32.CICS.CICSH###.DFHCSD)
  Accessmethod(Vsam)
  Action(              )
  Filecount(0001)
  Validity(Valid)
  Object(Base)
  Recovstatus(Notrecovable)
  Backuptype(Static)
  Frlog()
  Availability( Available )
  Lostlocks()
  Retlocks(Noretained)
  Uowaction(              )
  Basedsname(CICSTS32.CICS.CICSH###.DFHCSD)
  Quiescestate( Unquiesced )
  Fwdrecovlsn()
```

Figure 89. The expanded display of an individual entry

CEMT INQUIRE DSNAME



Displayed fields

Accessmethod
displays the access method of the data set. The values are:
Vsam Virtual storage access method (VSAM).
Bdam
Basic direct access method (BDAM).

Action (*input only field***)**
specifies the action to be taken on this data set. The values are:

Remove
This option is used when a data set is no longer required on the local system. It deletes the data set name block for the named data set. The REMOVE option can be used only when there are no file control resource definition entries that refer to the specified data set. If any file definitions refer to the data set name when you enter SET DSNAME REMOVE, you receive an error message indicating that the use count of the data set name block is not zero. To ensure that no file definitions refer to the data set name:

- Close all files that refer to the data set. These files can be found using the command:

```
CEMT INQUIRE FILE(*) DSNAME(dsname)
```

- Either remove the data set name from each file entry (by over-typing the data set name with blanks on the CEMT display), or discard the file definition entirely.

If you have a base data set with one or more alternate indexes, you are unable to remove the data set name block for the base until you have removed all references to the base data set and all references to paths.

Recovered

The data set has been restored from a backup version and forward recovery has run and completed successfully. CICS normally updates the backup-while-open (BWO) attributes for the base cluster data set BASEDSNAME, using MVS/DFP™ 3.2 Callable Services.

If the BWO attributes are not updated after the restore of a backup copy of a data set that was taken using BWO, CICS fails a subsequent file open because the data set is marked as down-level.

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.

RESETLOCKS applies only to shunted UOW log records that are classed as backout-failed and commit-failed. Therefore it cannot be issued for a BDAM data set. Backout-failed and commit-failed log records are not shunted for BDAM data sets.

You might 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.

Note:

1. This option does not apply to shunted *in-doubt* UOWs. 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 CEMT SET DSNAME command for information about these options).
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 Specifies that shunted UOW log records, caused by failed backout and commit processing as a result of the failure of this data set, should be retried. This is similar in operation to the EXEC CICS 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- 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.

Messages, issued at the time of a data set failure that causes UOWs to be shunted, recommend the actions that are required to recover from the failure.

RETRY does not affect data integrity, and can be used safely at any time to enable some failed recovery work to complete.

Availability
displays information 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. If the file is a BDAM data set, no information is returned. The values are:

Available
The data set is available for use according to the CICS data set name block. CICS can issue either RLS or 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.

Unavailable
The data set is marked by CICS as not available for use. The CICS region is unable to open the data set in either RLS or non-RLS mode.

AVAILABILITY can be reset by specifying AVAILABLE.

Backuptype
displays the backup type of the data set. The values are:

Static
The data set is accessed in non-RLS mode and is not eligible for BWO processing. All the files in this data set must be closed and the data set serialized by a backup manager such as DFSMS/MVS before a data set backup copy can be made.

Dynamic
The data set is accessed in RLS mode and is eligible for BWO processing. Data set backup is allowed while it is open for output, using a data set backup manager such as DFSMShsm or DFSMSdss.

Hierarchical storage manager (DFSMShsm) and data set services (DFSMSdss), which perform data set backup and recovery operations, are components of IBM Data Facility Storage Management Subsystem (DFSMS/MVS) program product.

Undetermined
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. NOTAPPLIC is also returned if the dataset has been opened in RLS mode. The VSAM catalog should be referred to get the BWO status.

Basedsname (*value*) (VSAM only)
When PATH is displayed as well, this is the name of the base data set with which this path is associated.

When BASE is displayed as well, this is the same as the name shown in the Dsn(*value*) field.

Dsname (*value*)
indicates that this panel relates to an external data set inquiry and displays the 44-character name of an external data set.

Filecount (*value*)
displays information on how many different files are currently associated with this data set. You can associate a file with a data set by a job control DD statement, or dynamically by a DSNAME parameter on the file definition. If a file has been associated by a job control DD statement, the file name cannot be changed by a CEDA command or by a CEMT or EXEC CICS SET FILE DSNAME command.

FILECOUNT does not show how many files are **open** against the data set; it only shows how many files are **associated** with the data set. If a file is associated with a data set and is then closed and associated with a different data set, the FILECOUNT of the first data set decreases by 1, and the FILECOUNT of the second data set increases by 1.

Frlog (*value*)
displays the numeric journal identifier of the journal being used to log the updates for forward recovery if this data set is forward recoverable (FWDRECOVABLE). The FRLOG value is undefined if the data set is not forward recoverable. You can define a data set as forward recoverable either in the ICF catalog or, if it is accessed in non-RLS mode, in the file resource definition. This option is meaningful only for data sets accessed in LSR or NSR mode, and for which the forward recovery attributes are obtained from the CICS file resource definition.

For data sets accessed in RLS mode (if DFSMS™ 1.3 is installed), or non-RLS mode data sets for which CICS obtains the recovery attributes from the ICF catalog, CICS displays a value of zero.

Fwdrecovlsn (*value*)
displays the name of the log stream that is used to log the updates if this data set is defined as forward recoverable. CICS returns nulls 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, 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.

Lostlocks
displays whether there are any lost locks for this data set. The possible values displayed are:

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. (The lost locks belong to a another (remote) CICS region.)

Recoverlocks
The data set has lost locks and therefore is unavailable. This CICS system is performing lost-locks recovery.

Object
displays the type of the data set. The values are:
Path (VSAM only)
This data set is a path leading to another data set.
Base The named data set is a base data set, as opposed to a path data set.

Quiescestate
displays the RLS quiesce state of the data set. The information is obtained from the ICF catalog entry for the data set. The values displayed are:
Quiesced
A data set has been quiesced. It means that RLS ACBs cannot open against the data set and no CICS region has an RLS ACB currently open against this data set. However, the data set can be opened in non-RLS mode.
Unquiesced
This is the normal value for a data set that is not quiescing or has not been quiesced. It indicates that RLS or non-RLS ACBs can be opened against the data set, the mode being established by the first open. After an ACB is opened in one mode, other ACBs can be opened only in the same mode.

If a data set is quiescing because a SET DSNAME QUIESCE command was issued in the same region as the CEMT I DSN command, the words “BEING QUIESCED” are displayed.

If a data set is quiescing, the words “BEING QUIESCED” are displayed for the CICS region in which the command was issued.

If quiesce does not apply (for example, for a BDAM data set), nothing is displayed.

Quiescestate can be reset by specifying Quiesced|Unquiesced or:
Immquiesced
Causes all existing CICS open RLS ACBs throughout the sysplex to be closed and the data set to be 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 ACBs, causing in-flight UOWs to back out. Any OUWs that fail backout are shunted. No RLS ACBs can open against this data set, but non-RLS open requests are permitted.

Recovstatus
displays the recovery status of the data set. The values are:
Fwdrecovable
All updates to the data set are logged for both backout and forward recovery.
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 or EXEC CICS SET DSNAME REMOVE command, or by an initial or cold start of CICS (if the associated data set is not in backout-failed state). 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 of the data set is not known because files accessing the data set have not been opened for output.

Retlocks
displays whether there are any retained record locks, as a result of deferred recovery work by this CICS region, for the specified data set.

Retained
This CICS region has deferred recovery work causing retained locks for the data set. One effect of this is that, if the data set was last opened in RLS mode, the locks are RLS locks and therefore the data set cannot be opened in non-RLS mode.

Another effect is that any FILE definitions that specify this data set cannot be changed to specify a different data set.

If the data set is a BDAM data set, or a VSAM data set accessed in non-RLS mode, the locks are CICS record locks, otherwise they are RLS record locks.

Noretained
This CICS region has no deferred recovery work for the base data set, and therefore no retained locks.

Note: **Noretained** can also mean that the CICS region is currently processing deferred recovery work. When the recovery work is successfully completed, the value remains as NORETAINED until more deferred work is created. If the recovery work is not successfully completed, the value reverts to RETAINED.

Uowaction (*input only field***)**
specifies the action to be taken for all shunted units of work that reference this data set. The values are:

Backout
Specifies that all shunted in-doubt units of work that hold locks on the data set should be backed out.

Commit
Specifies that all shunted in-doubt units of work that hold locks on the data set should be committed. Specifies that all shunted in-doubt units of work that hold locks on the data set should be forced to BACKOUT or COMMIT, as specified by the ACTION attribute on the transaction resource definition.

Validity
displays the validity of the data set name. The values are:

Valid The data set name has been validated against the VSAM catalog by opening a file against that data set.

Invalid
The data set name has not been validated against the VSAM catalog by opening a file against that data set.

CEMT INQUIRE DUMPDS

Retrieve information about the CICS dump data sets.

Description

INQUIRE DUMPDS tells you which dump data set (A or B) is currently open.

Instead of using the SET command, you can overtype your changes on the INQUIRE screen after tabbing to the appropriate field. See “Overtyping a display” on page 128.

For information on using dumps, see the *CICS Problem Determination Guide*.

Input

Press the Clear key and type `CEMT INQUIRE DUMPDS` (the minimum abbreviation is `CEMT I DU`).

You will get a display screen, a sample of which is shown below. If you want to change any of the values shown, you can then tab to the highlighted or blank fields and overtype them as required.

Sample screen

```
IN DUMPDS
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Dum Cur(A) Ope
```

Figure 90. CEMT INQUIRE DUMPDS screen

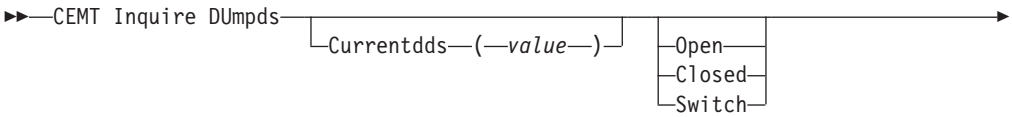
Note: There are blank fields on the screen where a value does not apply or is ‘negative’; that is, the value begins with ‘No’. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 91.

```
IN DUMPD
RESULT - OVERTYPE TO MODIFY
  Dumpds
  Currentdds(A)
  Openstatus( Open )
  Switchstatus( Noautoswitch )
```

Figure 91. The expanded display

CEMT INQUIRE DUMPDS



Displayed fields

Currentdds (*value*)
display a 1-character indicator as to which of the two dump data sets is active. This can be A or B.

DUmpds
indicates that this panel relates to a dump data set inquiry.

Openstatus
display the status of the active CICS dump data set. The values are:
Open The active CICS dump data set is open.
Closed
The active CICS dump data set is closed.
Switch
The currently active dump data set is to become inactive, and the currently inactive data set is to become active.

Note: You can reset this value by overtyping it with a different value.

Switchstatus
display whether there is to be an automatic switch to the inactive dump data set when the active dump data set is full. The values are:
Autoswitch
The switch occurs **only** when the active dump data set is full, but not subsequently.
Noautoswitch
When the active dump data set is full, there is not to be an automatic switch to the inactive dump data set.

Note: Because this is a 'negative' attribute, the field appears blank. You can, however, tab to this field and overtype it with a different value.

Note: You can reset this value by overtyping it with a different value.

CEMT INQUIRE ENQ

Retrieve information about enqueues held or waited on by a UOW, or about UOWs holding or waiting on a specified enqueue.

Description

INQUIRE ENQ returns information about the enqueues held by a UOW. 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.

You can use INQUIRE ENQ to indicate which records and queues would be affected if the UOW were forced.

The transaction also returns information about UOWs that are waiting on the enqueues. This enables you to diagnose enqueue deadlocks between tasks wanting to update the same resources.

Enqueues are typically held in active state, which means that other tasks are allowed to wait for the enqueue. However, if the UOW that owns the enqueue suffers an in-doubt failure, the enqueue is usually converted to the retained state until the in-doubt failure can be resolved. INQUIRE ENQ also retrieves information about these enqueues and can be used to identify which records and queues would be affected if the UOW were forced.

Note:

1. Both UOW-lifetime and task-lifetime enqueues are returned by INQUIRE ENQ. (For an explanation of UOW- and task-lifetime enqueues, see the MAXLIFETIME option of the EXEC CICS ENQ command.).
2. 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.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE ENQ` (the minimum abbreviation is `CEMT I ENQ`). You get a display that lists all enqueues for all the current UOWs.
- Type `CEMT INQUIRE ENQ` (`CEMT I ENQ`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i enq d`, the resulting display shows you the details only of enqueues on records in VSAM data sets open in non-RLS mode or CICS-maintained data tables.

Sample screen

```
INQ ENQ
STATUS:  RESULTS
 Enq(AC143001D4A72601) Tra(CECI) Tas(0000038) Act Exe Own
    Res(ACCOUNTS.DBASE1           ) Rle(015) Enq(00000000)
 Enq(AC143001D4A72601) Tra(CECI) Tas(0000038) Act Add Own
    Res(g...                      ) Rle(004) Enq(00000000)
```

Figure 92. CEMT INQUIRE ENQ screen

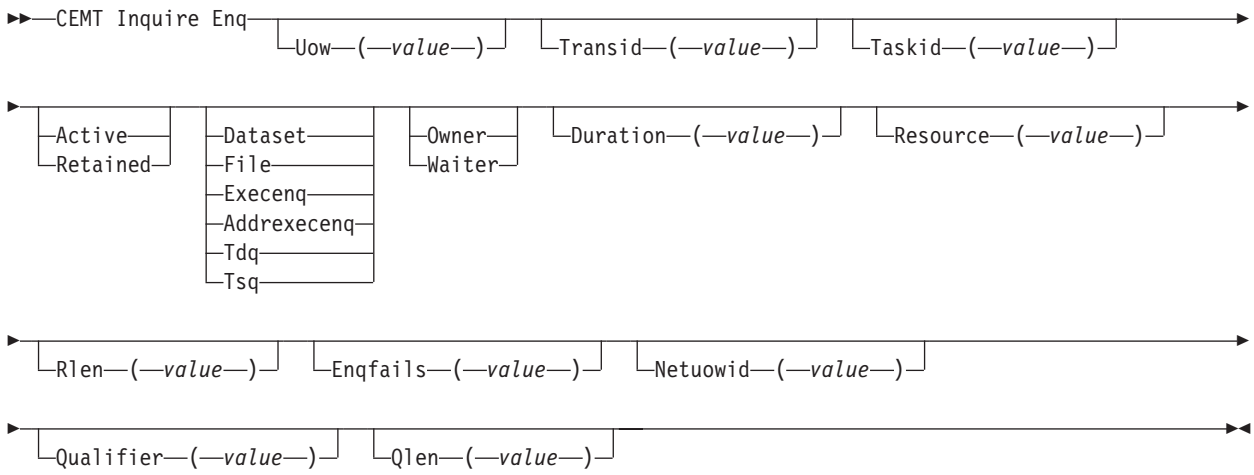
Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 93 on page 202.

```
INQ ENQ
RESULT
  Enq
  Enq(AC143001D4A72601)
  Transid(CECI)
  Taskid(0000038)
  State(Active)
  Type(Execenq)
  Relation(Owner)
  Duration()
  Resource(ACCOUNTS.DBASE1)
  Rlen(015)
  Enqfails(00000000)
  Netuowid(..GBIBMIYA.IYA2T5C2...Mx...)
  Qualifier()
  Qlen(000)
```

Figure 93. The expanded display of an individual entry

CEMT INQUIRE ENQ



Displayed fields

Duration (*value*)
displays the elapsed time (in seconds) since the enqueue ntered its current state of OWNER, WAITER or RETAINED.

Enqfails (*value*)
displays, for enqueues held in retained state, the number of failed enqueue attempts for this resource since the enqueue was last acquired. This indicates how many UOWs have received LOCKED responses due to this enqueue. The ENQFAILS option helps you identify which UOWs are causing “bottlenecks”.

For enqueues held in action state, ENQFAILS displays zero.

Enqscope (*value*)
If the enqueue has sysplex scope, ENQSCOPE displays 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 displays blanks.

All CICS systems with the same ENQSCOPE value share the same sysplex Enqueue name space.

Netuowid (*value*)
displays a 27-character string containing the network-wide LU6.2 name of the UOW that owns or is waiting for the enqueue.

Note: Nondisplayable characters appear as periods. You can use PF2 on the expanded panel: to see the value in hexadecimal.

Qualifier (*value*)
displays a 1- through 255-character qualifier (for example, record id) that further identifies the resource associated with the enqueue lock. The data displayed depends on the type of enqueue, as shown in Table 3 on page 204. If no QUALIFIER data applies (that is for EXECENQ, ADDREXECENQ, or TSQ), a value of zero is displayed.

Qlen (*value*)
displays the length of the data, in the range 0 through 255, displayed in the QUALIFIER field.

Relation
displays whether the data is associated with the owner of the enqueue or with a task waiting for the enqueue. The values are:

Owner
The (*value*), NETUOWID, TASKID, and TRANSID are those of the owner of the enqueue.

Waiter
The (*value*), NETUOWID, TASKID, and TRANSID are those of a waiter for the enqueue.

Resource (*value*)
displays a 1- through 255-character string containing the name of the resource associated with the enqueue lock. The data displayed depends on the type of enqueue, as shown in Table 3 on page 204.

Note: Nondisplayable characters appear as periods. You can use PF2 on the expanded panel: to see the value in hexadecimal.

Rlen (*value*)
displays the length of the data, in the range 0 through 255, displayed in the RESOURCE field.

State
displays the state that the enqueue is held in. The 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 (*value*)
displays the number of the task associated with the UOW. If the UOW is shunted, it is the task number associated with the UOW before it was shunted.

Transid (*value*)
displays the 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
displays the type of resource being enqueued upon. The values are:

Dataset
The resource is a record in a VSAM data set (or a CICS-maintained data table). RESOURCE contains the name of the data set and QUALIFIER contains the record identifier.

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.

Execenq
The resource is associated with an EXEC CICS ENQ request. RESOURCE contains the enqueue argument passed on the request.

Addrexecenq
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)

Tdq 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 a read or write lock is held for the queue.

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.

Tsq The resource is a recoverable temporary storage queue. RESOURCE contains the name of the queue.

The data displayed in the RESOURCE and QUALIFIER fields depends on the resource type, as shown in Table 3.

Table 3. Data displayed in RESOURCE and QUALIFIER

Type	Resource	Qualifier
DATASET	Dataset name	Record identifier
EXECENQ	EXEC enqueue argument	None
ADDREXECENQ	Address of EXEC enqueue argument	None
FILE	File name	Record identifier
TDQ	TD queue name	FROMQ or TOQ
TSQ	TS queue name	None

Uow (*value***)**
displays the 16-character local identifier of the associated unit of work.

CEMT INQUIRE ENQMODEL

Retrieve information about enqmodel definitions on the local system.

Input

Press the Clear key and type `CEMT INQUIRE ENQMODEL` (the minimum abbreviation is `CEMT I ENQM`).

You get a display screen, a sample of which is shown below. If you want to change any of the values shown, you can then tab to the highlighted or blank fields and overtype them as required.

Sample screen

```
 INQ ENQMODEL
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Enqm(*       )                                          NOT FOUND
```

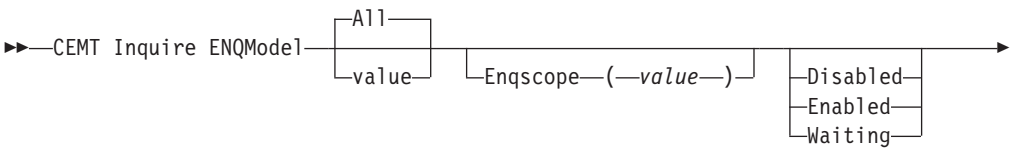
Figure 94. CEMT INQUIRE ENQMODEL screen

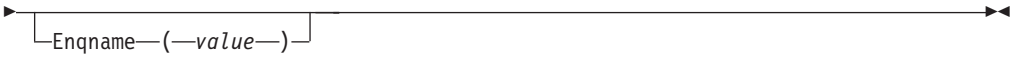
```
i enqm all
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Enqm(fred    ) Enqs(TEST) Ena
    Enqn(HURSLEY                  )
 Enqm(norman  ) Enqs(SYSP) Ena
    Enqn(NORMAN                   )









                                        SYSID=HT61 APPLID=CICSHT61
 RESPONSE: NORMAL                     TIME:  13.23.25  DATE: 02.21.01
PF 1 HELP       3 END       5 VAR        7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 95. CEMT INQUIRE ENQMODEL expanded display screen

Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

CEMT INQUIRE ENQMODEL





Displayed fields

All
is the default.

ENQModel
The 8-character identifier of the resource definition

Enqname
The 1 - 255 character resource or generic name.

EXEC ENQ requests 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 to be sysplex-wide, qualified by the 4-character enqscope.

Enqscope
Returns the 4-character name used to qualify SYSPLEX ENQUEUE requests issued by this CICS region. Four blanks indicates that the ENQ is LOCAL.

Status
displays the treatment of matching ENQ requests. The values are:

Disabled
Matching ENQ requests are rejected, and the issuing task abended abcode ENQ_DISABLED. Matching INSTALL/CREATE/DISCARD requests are processed.

Enabled
Matching ENQ requests are processed in the normal way.

Waiting
Matching ENQ requests are rejected, and the issuing task abended abcode ENQ_DISABLED. There are INSTALL/CREATE/DISCARD requests waiting to be processed.

CEMT INQUIRE EXCI

Retrieve information about jobs using the external CICS interface.

Description

INQUIRE EXCI identifies the names of batch jobs currently connected to CICS through the interregion communication (IRC) facility. These jobs are using the external CICS interface (EXCI) to link to a server program running in CICS.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE EXCI` (the minimum abbreviation is `CEMT I E`). You get a display that lists the current status.
- Type `CEMT INQUIRE EXCI` (`CEMT I E`) followed by `Task(number)` to limit the scope of the inquiry.

Sample screen

```
INQ EXCI
STATUS:  RESULTS
 Exc(HADBS1..CICSWK1 - MV29            ) Tas(0000121)
 Exc(AHCLN2..TRNSL 1 - MV29            ) Tas(0000125)
    Uri(B 67B1F77EB7D       21 1 1     )
```

Figure 96. CEMT INQUIRE EXCI screen

Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 97.

```
INQ EXCI
RESULT
    Exc(HADBS1..CICSWK1 - MV29           ) Tas(0000121)
    Task(0000121)
```

Figure 97. The expanded display of an individual entry

CEMT INQUIRE EXCI

Displayed fields

EXci (*value***)**
indicates that this panel relates to an EXCI inquiry.

The value consists of:

```
jobname.stepname.procname - mvsid
```

One or both of the stepname and procname may not be present, indicated by the periods (.) being adjacent to one another.

The mvsid identifies the MVS system on which the job is running. If XCF/MRO is in use, the job can reside on a different MVS image from that on which CICS is running.

Task (*value***)**
displays the 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 (*value***)**
displays, 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.

CEMT INQUIRE FEPI

Comments

For information on CEMT INQUIRE FECONNECTION, CEMT INQUIRE FEPOOL, CEMT INQUIRE FEPROPSET, and CEMT INQUIRE FETARGET, see the *CICS Front End Programming Interface User's Guide*.

CEMT INQUIRE FILE

Retrieve information about VSAM, BDAM, and remote files, and files that refer to data tables.

Description

The values that are returned to you can vary according to when the command is issued. For example, if the file is closed when you issue the command, much of the information you receive tells you the state of the file when it is next opened. If the file has never been opened, you receive default or null values for some of the options, which could change when the file is opened.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE FILE` (the minimum abbreviation is `CEMT I FI`). You get a display that lists the current status of all files installed.
- Type `CEMT INQUIRE FILE` (`CEMT I FI`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i fi v op en up`, the resulting display will show you the details of only those files that are VSAM, open, enabled, and updatable.

You can then tab to the highlighted or blank fields and overtype them with the required values.

(*value***)**
specifies one or more names (1-8 characters) of files defined to CICS.

ALl
is the default. Information about all files is given, unless you specify a selection of files to be queried.

Sample screen

```
IN FILE
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Fil(DFHCMACD) Vsa Ope Ena Rea                      Sha
       Dsn( CICSTS32.CICS.DFHCMACD                          )
 Fil(DFHCSD  ) Vsa Clo Une Rea Upd Add Bro Del     Sha
       Dsn( CICSTS32.CICS.CICSH.DFHCSD                      )
 Fil(DFHDBFK ) Vsa Clo Ena Rea Upd Add Bro Del     Sha
 Fil(DFHLRQ  ) Vsa Ope Ena Rea Upd Add Bro Del     Sha
       Dsn( CICSTS32.CICS.CICSH.DFHLRQ                      )
 Fil(FILEA   ) Vsa Clo Ena Rea Upd Add Bro Del     Sha
       Dsn( CICSTS32.CICS.CICSH.FILEA                       ) Cft Loa
```

Figure 98. CEMT INQUIRE FILE screen

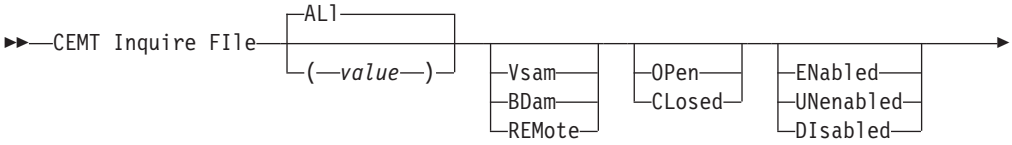
Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

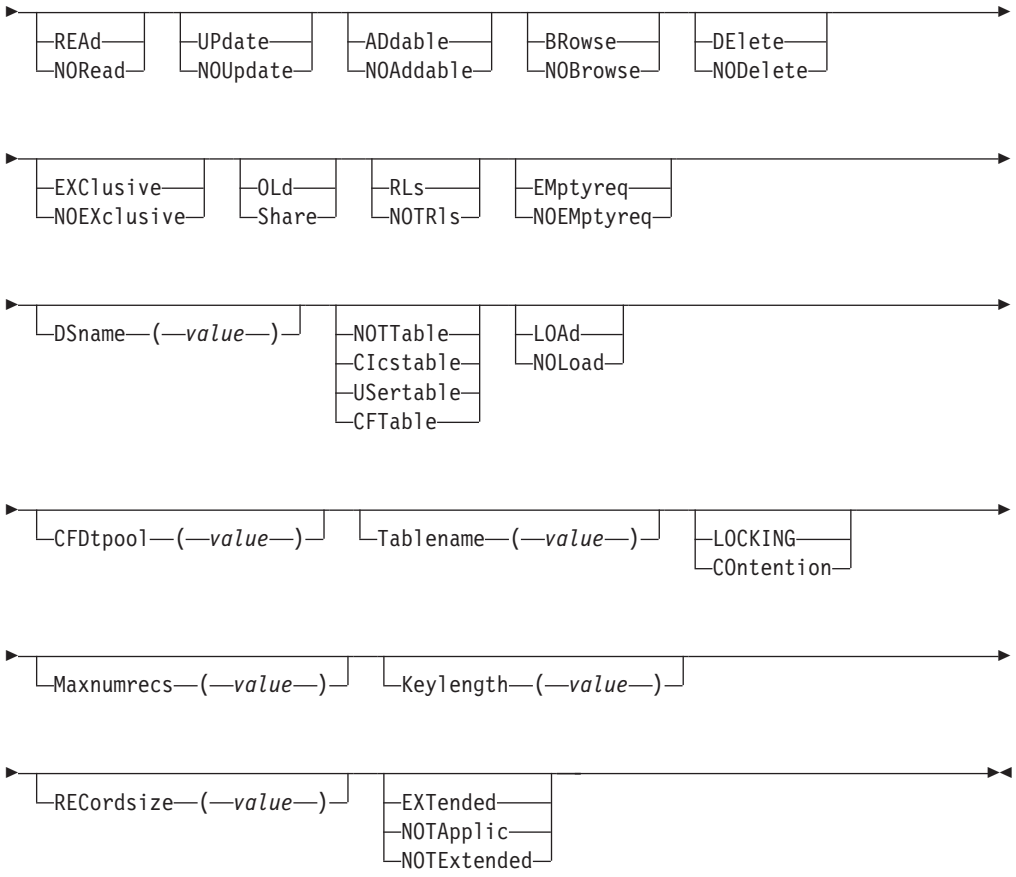
If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 99.

```
 INQ FILE
 RESULT - OVERTYPE TO MODIFY
   File(DFHCSD)
   Accessmethod(Vsam)
   Openstatus( Closed )
   Enablestatus( Unenabled )
   Readstatus( Read )
   Updatestatus( Update )
   Addstatus( Addable )
   Browsestatus( Browse )
   Deletestatus( Delete )
   Exclstatus(              )
   Disposition( Share )
   Rlsaccess( Notrls )
   Emptystatus( Noemptyreq )
   Dsname( CICSTS32.CICS.CICSH.DFHCSD )
   Table( Nottable )
   Loadtype( Noload )
   Cfdtpool(          )
+  Tablename(          )
   Updatemodel(         )
   Maxnumrecs( 00000000 )
   Keylength( 000 )
   Recordsize( 00000 )
   Rbatype( Notapplic )
```

Figure 99. The expanded display of an individual entry

CEMT INQUIRE FILE





Displayed fields

Accessmethod
displays the access method of the file. The values are:
Vsam Virtual storage access method (VSAM).
Bdam
Basic direct access method (BDAM).
Remote
The file is defined as remote.

Addstatus
displays whether new records can be added to the file. The values are:

Addable
New records can be added to the file.

Noaddable
New records cannot be added to the file.

Note: You can reset this value by overtyping it with a different value.

Browsestatus
displays whether you can browse the file. The values are:
Browse
You can browse records in the file.
Nobrowse
You cannot browse records in the file.

Note: You can reset this value by overtyping it with a different value.

Cfdtpool (*name***)**
displays, for a file that refers to a coupling facility data table, the 8-character name of the coupling facility data table pool in which the CFDT resides.

If the pool name is not defined, the name is displayed as 8 blanks. If the pool name is specified, it is always displayed even if the file definition does not refer to a coupling facility data table.

Deletestatus
displays whether you can delete records from the file. The values are:

Delete
You can delete records from the file.

Nodelete
You cannot delete records from the file.

Note: You can reset this value by overtyping it with a different value.

Disposition
displays the disposition of this file. The values are:

Old The disposition of the data set associated with the file is defined as unshared for dynamic allocation.

Share
The disposition of the data set associated with the file is defined as shared for dynamic allocation.

Note: You can reset this value by overtyping it with a different value.

Dsname (*value***)**
displays the name (1–44 characters) of the VSAM or BDAM data set with which this file is associated.

For a coupling facility data table, CICS displays the name of the source data set if the file specifies that the table is initially loaded from a data set, otherwise CICS displays blanks.

CICS also displays blanks for files that have their DSNAME specified on a DD statement in the CICS JCL if the file has not been opened since the last cold or initial start.

Note: If the file is disabled, you can change this name by overtyping it with a different value.

Emptystatus (*VSAM only***)**
displays whether the data set is to be made empty when a file that references it is next opened. This is valid only for data sets that have been defined as reusable and are accessed in non-RLS mode. The values are:

Emptyreq
Applies only to a VSAM file that has been defined to VSAM as a cluster with the REUSE parameter. It indicates that when a file is next opened, its data is erased.

Noemptyreq
Applies only to VSAM files. It indicates that when a file is next opened, its data is not erased.

Note: You can reset this value by overtyping it with a different value.

Enablestatus
displays whether transactions can access the file. The values are:

Enabled
The file is available for use by transactions and, if closed, it is opened on the first request.

Unenabled
The file is not available for use by transactions except for those that are currently using it. If there are any such users, 'BEING CLOSED' is also displayed.

This status is the same as DISABLED except that it occurs implicitly when a SET FILE CLOSE is requested. The file is enabled implicitly by a SET FILE OPEN command.

Disabled
The file is not available for use by transactions except for those that are currently using it. If there are any such users, 'BEING DISABLED' is also displayed.

The file can be reenabled by a SET FILE ENABLED command. (See also UNENABLED.)

It is not possible to disable a remote file.

Note: You can reset this value by overtyping it with Enabled or Disabled.

Exclstatus (*BDAM only*)
displays whether records on this file are to be placed under exclusive control when a read for update is issued. The values are:

Exclusive
Records on the file are protected from simultaneous update.

Noexclusive
Records on the file are not protected from simultaneous update.

File (*value*)
indicates that this panel relates to a file inquiry and displays an 8-character file name.

Keylength (*value*)
displays 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. The KEYLENGTH attribute is valid only for remote files and coupling facility data tables. CICS returns zero for local files which are closed.

Note:

1. If the file is closed and the key length is not defined in the file definition, CICS displays a zero value.
2. If the file is closed and a key length is defined on the file definition, CICS displays the value from the file definition.
3. Generally, if a file is open, CICS obtains the key length from the associated data set, in which case CICS displays 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 defined value when the file is open. This value must also match that of the coupling facility data table if it has already been created.

Loadtype
displays the load type for a coupling facility data table. The 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.

Maxnumrecs (*value*)
displays the maximum number of records that the data table for this file can hold.

Note:

1. If the file resource definition specifies a MAXNUMRECS numeric value, even though the object is not a table (NOTTABLE is displayed on the TABLE option), CICS displays the specified value. The value can be in the range 1 through 99 999 999.
2. If the file resource definition is specified with MAXNUMRECS(NOLIMIT), meaning the number of records is unlimited, CICS displays a value of zero. (Internally, CICS holds NOLIMIT as the maximum positive fullword value (+2147483647 or X'7FFFFFFF').
3. If the file is remote, CICS displays a value of minus 1 (-1).
4. 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 displays the value in the file definition until the file is opened, after which CICS displays 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 displays the new value only after a file control operation on the table. Until then, CICS continues to display the old value.
   - You can use the server DISPLAY TABLE console command to obtain the current value for a coupling facility data table.

Note: You can reset this value by overtyping it with a different value. The new value takes effect only when a new data table is created (which happens on each file open for a CICS-maintained or user-maintained data table, but only when the table does not already exist for a CF data table).

Openstatus
displays the open status of the file. The values are:

Open The file is open. If the file is OPEN ENABLED, it is available for data accesses by CICS transactions. If it is OPEN DISABLED, it must first be enabled before it is available.

Closed
The file is closed; the access method does not allow accesses to the data until the file is opened. The file can be opened either explicitly by the user or, if the file has the ENABLED attribute, implicitly by CICS on the next reference to the file.

A file with the DISABLED attribute can be enabled only by a SET FILE ENABLED command. A file with the UNENABLED attribute can be enabled by a SET FILE ENABLED command or SET FILE OPEN command.

The CLOSED status may be a temporary status for a file that has just come out of LOADMODE status.

Note: You can reset this value by overtyping it with one of the above Openstatus values, or with:

Forceclose
Specifies that the file is to be closed, and the tasks of any existing users of the file are to be abended.

Rbatype
displays whether, for VSAM files, the data set uses extended addressing. The values are:

Extended
This VSAM data set uses extended relative byte addressing and therefore can hold more than 4 gigabytes of data.

Notapplic
One of the following is true:
- The data set is BDAM.
- The file is remote.
- The file is not open.

Notextended
This VSAM data set does not use extended relative byte addressing and therefore cannot hold more than 4 gigabytes of data.

You cannot modify the contents of this field.

Readstatus
displays whether you can read records from the file. The values are:

Read You can read records in the file.

Noread
You cannot read records in the file.

Note: You can reset this value by overtyping it with a different value.

Recordsize (*value*)
displays the actual size of fixed-length records, or the maximum size of variable-length records.

If the file is not open, CICS displays the value specified in the installed file definition. If the file definition does not specify a record size, CICS displays a zero value.

Generally, if a file is open, CICS obtains the record size from the associated data set, in which case CICS displays 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 defined value when the file is open. This value must also match that of the coupling facility data table if it has already been created.

Rlsaccess (*VSAM only*)
displays whether the file is defined to be accessed in RLS mode. The values are:

Rls The file is defined to be accessed in RLS mode.

Notrls
The file is defined to be accessed in a non-RLS mode (LSR or NSR)

Note: You can reset this value by overtyping it with a different value.

Table
displays whether the file name specified on the FILE parameter represents a data table. The values are:
Nottable
The file name does not represent a data table.
Cftable
The file name represents a coupling facility data table.
Cicstable
The file name represents a CICS-maintained data table.
Usertable
The file name represents a user-maintained data table.

Note: You can reset this value by overtyping it with a different value.

Tablename (*value*)
displays the table name specified for the coupling facility data table on the file resource definition, if one is specified, or displays the file name if no table name was specified on the file resource definition.

CICS displays blanks if the file does not refer to a coupling facility data table

Updatemodel (CFDT only)
displays the update model specified for the coupling facility data table in the installed file definition. The 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.

Updatestatus
displays whether the file is updatable. The values are:
Update
You can update records in the file.
Noupdate
You cannot update records in the file.

Note: You can reset this value by overtyping it with a different value.

CEMT INQUIRE GTFTRACE

Show the GTF status.

Description

CEMT INQUIRE GTFTRACE tells you whether CICS tracing to the MVS Generalized Tracing Facility (GTF) is started or stopped.

This command is intended for use at terminals that do not support the CETR full-screen tracing transaction.

For background information about GTF tracing, see the *CICS Problem Determination Guide*.

Input

Press the Clear key and type `CEMT INQUIRE GTFTRACE` (the minimum abbreviation is `CEMT I G`).

You will get a display screen, a sample of which is shown below. If you want to change the displayed value, you can then tab to the highlighted field and overtype it as required.

Sample screen

```
IN GTFTRA
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Gtf Sto
```

Figure 100. CEMT INQUIRE GTFTRACE screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 101.

```
IN GTFT
RESULT - OVERTYPE TO MODIFY
  Gtftrace
  Status( Stop )
```

Figure 101. The expanded display

CEMT INQUIRE GTFTRACE

Displayed fields

Gtftrace
indicates that this panel relates to a GTFTRACE inquiry.

Status
displays the status of CICS tracing to the MVS Generalized Tracing Facility (GTF). The values are:

Start CICS tracing to the MVS Generalized Tracing Facility (GTF) is in progress.

Stop CICS tracing to the MVS Generalized Tracing Facility (GTF) is not in progress.

Note: You can reset this value by overtyping it with a different value.

CEMT INQUIRE HOST

Inquire on the status of a virtual host.

Description

CEMT INQUIRE HOST allows you to retrieve information about a particular virtual host in the local CICS region.

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.

Syntax

Press the Clear key and type `CEMT INQUIRE HOST` (the minimum abbreviation is `CEMT I H`).

You will get a display screen, a sample of which is shown below. If you want to change the displayed value, you can then tab to the highlighted field and overtype it as required.

Sample screen

```
IN HOST
STATUS:  RESULTS - OVERTYPE TO MODIFY
  Hos(www.hursleytest.ibm.com   )              Ena
  Hos(www.hursleytest2.ibm.com  )              Ena
```

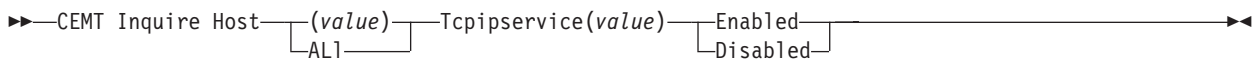
Figure 102. CEMT INQUIRE HOST screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 103.

```
IN HOST
RESULT - OVERTYPE TO MODIFY
    Host(www.hursleytest.ibm.com)
    Tcpipservice(TEST2)
    Enablestatus( Enabled )
```

Figure 103. The expanded display

CEMT INQUIRE HOST



Options

(*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 same name. A host name in a URIMAP definition can be up to 120 characters.

Tcpipservice *(value)*
displays the 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.

Enablestatus
displays the status of the virtual host. The possible values are:

Enabled
The URIMAP definitions that make up the virtual host can be accessed.

Disabled
The URIMAP definitions that make up the virtual host cannot be accessed.

CEMT INQUIRE INTTRACE

Show the status of CICS internal tracing.

Description

INQUIRE INTTRACE tells you whether CICS internal tracing is in progress.

This command is intended for use at terminals that do not support the full-screen tracing transaction CETR.

Input

Press the Clear key and type `CEMT INQUIRE INTTRACE` (the minimum abbreviation is `CEMT I IN`).

You will get a display screen, a sample of which is shown below. If you want to change the displayed value, you can then tab to the highlighted field and overtype it as required.

Sample screen

```
 IN INTTRA
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Int Sta
```

Figure 104. CEMT INQUIRE INTTRACE screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 105.

```
IN INTT
RESULT - OVERTYPE TO MODIFY
  Inttrace
  Status( Start )
```

Figure 105. The expanded display

CEMT INQUIRE INTTRACE

Displayed fields

Inttrace
indicates that this panel relates to an INTTRACE inquiry.

Status
displays the status of CICS tracing to the MVS Generalized Tracing Facility (GTF). The values are:
Start CICS internal tracing is in progress.
Stop CICS internal tracing is not in progress.

Note: You can reset this value by overtyping it with a different value.

CEMT INQUIRE IPCONN

Function

Retrieve information about IPCONNs.

Description

CEMT INQUIRE IPCONN returns information about the status of IPCONNs to a remote system or to another CICS region.

An IPCONN (or IPIC connection) is a TCP/IP communication link from your local CICS region to another CICS region, or another system.

You can reset the options either by typing the SET command or by overtyping at the appropriate place on the INQUIRE screen.

If you want to install a new IPCONN definition when one is already installed, you must set the connection OUTSERVICE before using the CEDA INSTALL command for your new definition.

Note: CEMT INQUIRE IPCONN returns information about IPIC connections. See also "CEMT INQUIRE CONNECTION" on page 154. The CEMT INQUIRE CONNECTION command returns information about MRO and ISC over SNA connections.

For information about the different kinds of intercommunication connections, see the *CICS Intercommunication Guide*.

Input

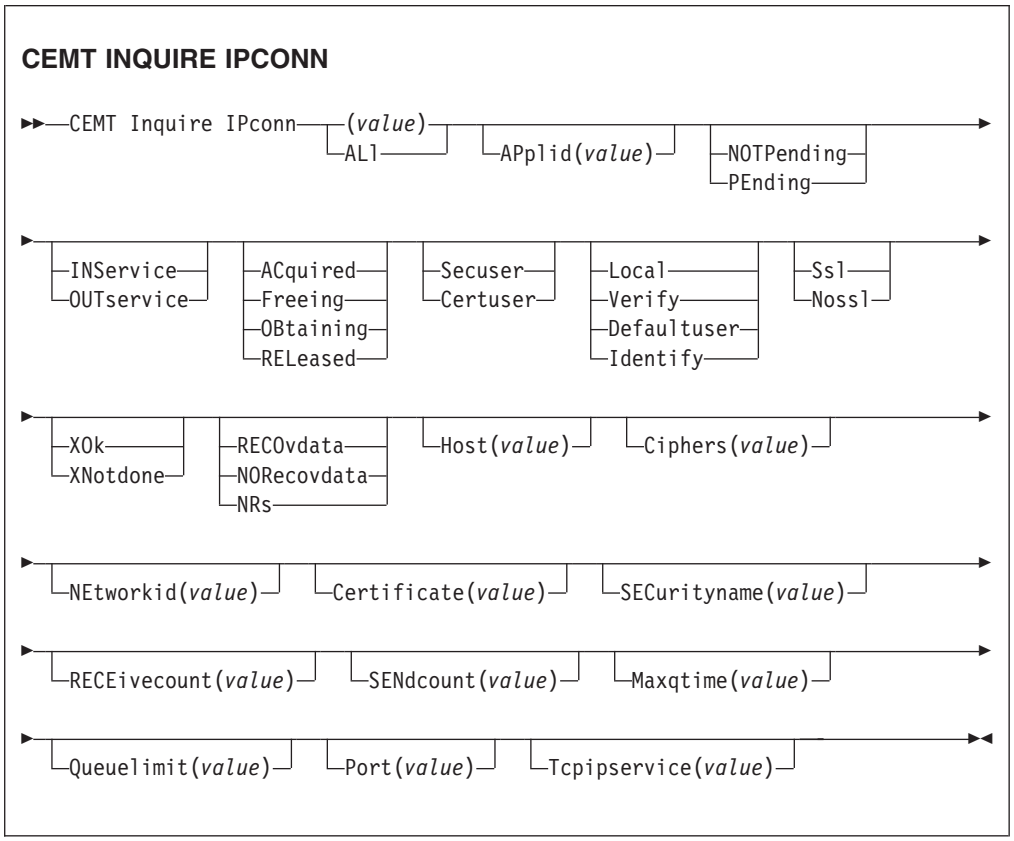
Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE IPCONN` (the minimum abbreviation is `CEMT I IP`). You get a display that lists the current status.
- Type `CEMT INQUIRE IPCONN` (`CEMT I IP`) followed by as many of the other attributes as are necessary to limit the range of information that you require.

You can then tab to the highlighted or blank fields and overtype them with the required values.

(*value***)**
specifies one or more names (1–8 characters) defined for an IPCONN.

All
is the default.



Displayed fields

Applid(*value*)
displays the name by which the remote system is known to the network (taken from the APPLID option of the IPCONN definition). This is the application identifier (*applid*) of the remote system, as specified on the APPLID option of its system initialization table. For XRF systems it is the generic applid.

Connstatus
indicates the state of the IPCONN between CICS and the remote system. The values are:

Acquired
The IPCONN is acquired. The criteria for ACQUIRED are:
- The capabilities exchange is complete.

Freeing
The IPCONN is being released.

Obtaining
The IPCONN is being acquired. The IPCONN remains in the OBTAINING state until all the criteria for ACQUIRED have been met.

Released
The IPCONN 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 IPCONN definition exists on the remote system and autoinstall was not active or not successful.
- The IPCONN on the remote system has been set out of service.
- AUTOCONNECT(NO) has been specified on the IPCONN definition.
- The IPCONN had been acquired but has since been released by an explicit operator command.

Certificate(*value*)
displays a 32-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 IPCONN connections. If the label is blank, the certificate nominated as the default for the key ring is used.

Ciphers(*value*)
displays 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*.

Host(*value*)
displays the host name of the remote system (for example, `abc.example.com`), or its dotted decimal IP address (for example, `9.20.181.3`).

Linkauth
displays a CVDA value that specifies how the user ID for link security is established in a CICS system with security initialized (SEC=YES).

CERTUSER
TCP/IP communication with the partner system must be configured for SSL and a certificate must be received from the partner system during SSL handshake.

The IPCONN must refer to a TCPIPSERVICE that is defined with SSL(CLIENTAUTH).

The received certificate must be defined to the external security manager so that it is associated with a user ID, which is used to establish link security.

SECUSER
Specifies that the user ID specified in SECURITYNAME is used to establish link security.

Maxqtime(*value*)
displays the maximum time, in seconds, for which allocate requests may be queued. The value is in the range 0-9999, or will have the standard null value of -1 if MAXQTIME(NO) is specified on the IPCONN definition.

Networkid
displays the network ID of the remote system. The value retruned is an 8-byte character string, which is is the value of the NETWORKID option of the IPCONN definition. If NETWORKID is not specified on the IPCONN definition, the value returned is the VTAM NETID or, for VTAM=NO systems, the value of the UOWNETQL system initialization parameter, of this CICS (that is, the CICS on which the IPCONN definition is installed).

The `Networkid` value is used in combination with the `Applid` value to ensure unique naming for connecting systems.

Pendstatus
displays whether there are any pending units of work for this IPCONN. The values are:

Notpending
There has been no mismatch of lognames with the partner.

Pending
There is resynchronization work outstanding for the connection but the partner system has performed an initial start, preventing completion of the resynchronization process. You can use the SET IPCONN NOTPENDING command to unilaterally commit or back out the units of work associated with the connection, according to their associated transaction definitions. You can also investigate the units of work individually and force them to commit or back out, in which case you must also complete the recovery activity by using a SET IPCONN NOTPENDING command to clear the PENDING condition.

If this is a CICS-to-CICS IPCONN, no new syncpoint work (that is, work involving sync level 2 protocols) can be transmitted across the connection until a SET IPCONN NOTPENDING command has been issued.

If you are not concerned by the loss of synchronization caused by the initial (or cold) start of the partner, you can cause the SET IPCONN NOTPENDING command to be issued automatically by specifying XLNACTION(FORCE) on the IPCONN definition.

For further information about pending units of work, see the *CICS Intercommunication Guide*.

Port(*value***)**
displays the port number to be used for outbound requests on this connection; that is, the number of the port on which the remote system will be listening. This may be set to NOTAPPLIC if the connection is never used for outbound traffic, as is usually the case for autoinstalled IPCONNs.

Queuelimit(*value***)**
displays the maximum number of allocate requests that can be queued for this connection. The value is in the range 0-9999, or will have the standard null value of -1 if QUEUELIMIT(NO) is specified on the IPCONN definition.

Receivecount(*value***)**
displays the number of RECEIVE sessions defined for this connection.

Recovstatus
indicates whether there is resynchronization work outstanding for the IPCONN. The IPCONN may never have been connected, have been quiesced and all resynchronization work completed, or disrupted without quiesce: in which case resynchronization may be necessary. The values are:

Norecovdata
Neither side has recovery information outstanding.

Nrs CICS does not have recovery outstanding for the IPCONN, but the partner may have.

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

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

Securityname(*value***)**
displays the security name of the remote system.

In a CICS system with security initialized (SEC=YES), the security name is used to establish the authority of the remote system.

The security name must be a valid RACF user ID on this region. Access to protected resources on this region is based on the RACF user profile and its group membership.

Sendcount(*value***)**
displays the number of SEND sessions defined for this IPCONN.

Servstatus
indicates whether data can be sent and received on the IPCONN. The values are:

Inservice
Data can be sent and received.

Outservice
Data cannot be sent or received.

Ssltype
displays a CVDA value specifying the level of secure sockets support being used for this service. CVDA values are:

NOSSL
The Secure Sockets Layer is not being used for this service.

SSL The Secure Sockets Layer is being used for this service.

Tcpipservice(*value***)**
displays the name of a PROTOCOL(IPIC) TCPIPSERVICE definition that defines the attributes of the inbound processing for this IPCONN.

Userauth
displays a CVDA value that specifies how the user ID for attach-time user security is established in a CICS system with security initialized (SEC=YES)

DEFAULTUSER
CICS does not accept a user ID and password from the partner system. All requests run under the default user ID.

IDENTIFY
Incoming attach requests must specify a user identifier but not a password.

LOCAL
CICS does not require a user ID or password from clients. All requests will run under the link user ID.

VERIFY
Incoming attach requests must specify a user identifier and a user password.

CEMT INQUIRE IRBATCH

Retrieve information about jobs using the external CICS interface.

Description

CEMT INQUIRE IRBATCH is retained for compatibility. You should use CEMT INQUIRE EXCI.

CEMT INQUIRE IRC

Show the IRC status.

Description

INQUIRE IRC tells you whether the interregion communication (IRC) facility is open or closed.

Input

Press the Clear key and type `CEMT INQUIRE IRC` (the minimum abbreviation is `CEMT I IRC`).

You will get a display screen, a sample of which is shown below.

Sample screen

```
 IN IRC
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Irc Clo DFHIR000
```

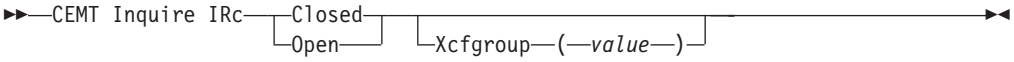
Figure 106. CEMT INQUIRE IRC screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 107.

```
IN IRC
RESULT - OVERTYPE TO MODIFY
  Irc
  Openstatus( Closed )
  Xcfgroup( DFHIR000 )
  Return(00000000)
  Reason(00000000)
```

Figure 107. The expanded display

CEMT INQUIRE IRC



Displayed fields

IRc
indicates that this panel relates to an IRC inquiry.

Openstatus (*value*)
displays whether the interregion communication (IRC) facility is open or closed. The values are:

Closed
Interregion communication for this CICS system is terminated after tasks using the session have terminated normally.

Open Interregion communication for this CICS system is initialized. This allows connections to be established between this system and other systems. You can control the availability of individual connections with the INSERVICE or OUTSERVICE settings on the connection names, using CEMT INQUIRE CONNECTION or CEMT SET CONNECTION.

The DFHIR3780 message is issued to the console log if the region made an unsuccessful attempt to start IRC.

Note:

1. You can reset this value by overtyping it with a different value, or with:
   Immclose
   All tasks using IRC sessions are to be abended when they next try to use the session. Interregion communication is terminated when all tasks that were using IRC have been abended or have terminated normally.
2. The values of Return and Reason will be nonzero only after certain SET (or overtype) failures. These are accompanied by a message number, which you should refer to in the*CICS Messages and Codes* manual.

Xcfgroup
displays the name of the cross-system coupling facility (XCF) group of which this region is a member.

If this region is not a member of an XCF group (because it has not signed on to IRC), XCFGROUP displays the XCF group the region would be in if XCF were opened.

For introductory information about XCF/MRO, see the *CICS Intercommunication Guide*.

CEMT INQUIRE JMODEL

Retrieve information about journal models and log stream names.

Description

The INQUIRE JMODEL command enables you to inquire on installed journal models to obtain corresponding log stream names.

Input

Press the CLEAR key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE JMODEL` (the minimum abbreviation is `CEMT I JM`). You get a display that lists the current status.
- Type `CEMT INQUIRE JMODEL` (`CEMT I JM`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So,

for example, if you enter `cemt i jm sm`, the resulting display will show you the details of only those journal models that are of SMF log stream type.

(*value***)**
specifies one or more journal models.

All
is the default.

Sample screen

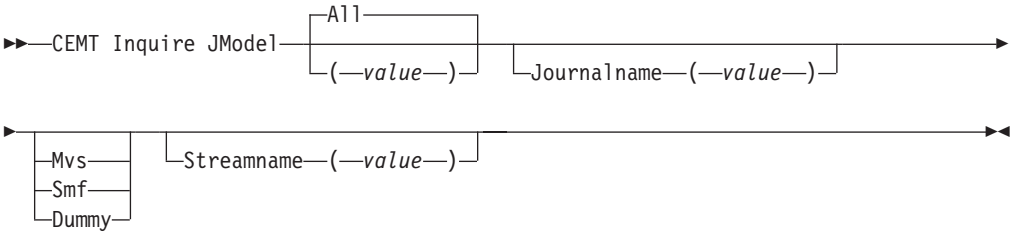
```
IN JMODE
STATUS:  RESULTS
 Jmo(DFHJ02  ) Jou(DFHJ02  ) Mvs
    Str(BELL.TESTLOG.FWDRECOV     )
 Jmo(DFHLOG  ) Jou(DFHLOG  ) Mvs
    Str(&USERID..&APPLID..&JNAME. )
 Jmo(DFHSHUNT) Jou(DFHSHUNT) Mvs
    Str(&USERID..&APPLID..&JNAME. )
```

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 108.

```
IN JMODE
RESULT
  Jmodel(DFHJ02)
  Journalname(DFHJ02)
  Type(Mvs)
  Streamname(BELL.TESTLOG.FWDRECOV)
```

Figure 108. The expanded display of an individual entry

CEMT INQUIRE JMODEL



Displayed fields

Jmodel (*value***)**
indicates that this panel relates to a journal model inquiry and displays the 8–character name of an installed journal model.

Journalname
displays the 8–character journal name, which can be a specific or a generic name, for which this journal model can be used.

Streamname
displays the MVS log stream name (LSN) associated with this journal model. The name can be a specific LSN, or a template using any 3 of the 4 symbols &USERID, &APPLID, &JNAME and &SYSID.

An LSN or template can be up to 26 characters in length.

Type
displays the log stream type. The possible types are:

Mvs Records for journals defined with this journal model are written to an MVS log stream.

Smf Records for journals defined with this journal model are written to the MVS SMF data set.

Dummy
Records for journals defined with this journal model are not written to any log stream.

For information on changing JMODEL definitions, see the *CICS Resource Definition Guide*.

CEMT INQUIRE JOURNALNAME

Show the status of the CICS system log and general logs.

Description

The INQUIRE JOURNALNAME command enables you to inquire on the status of the CICS system log and general logs. It replaces the INQUIRE JOURNALNUM command.

Input

Press the CLEAR key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE JOURNALNAME` (the minimum abbreviation is `CEMT I JO`). You get a display that lists the current status.
- Type `CEMT INQUIRE JOURNALNAME` (`CEMT I JO`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i jo sm`, the resulting display will show you the details of only those logs that are of SMF log stream type.

(*value***)**
specifies one or more journal names.

To inquire on journals defined with a numeric identifier in the range 1–99, specify journal name DFHJ*nn*, where *nn* is the journal number.

To inquire on the system log, specify DFHLOG.

To inquire on the log-of-logs, specify DFHLGLOG.

All
is the default.

Sample screen

```
IN JOUR
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Journaln(DFHJ02  ) Smf Ena
 Journaln(DFHLGLOG) Mvs Ena
    Str(CICSHT##.CICSHTA1.DFHLGLOG  )
 Journaln(DFHLOG  ) Mvs Ena
    Str(CICSHT##.CICSHTA1.DFHLOG    )
 Journaln(DFHSHUNT) Mvs Ena
    Str(CICSHT##.CICSHTA1.DFHSHUNT  )
```

Figure 109. CEMT INQUIRE JOURNALNAME screen

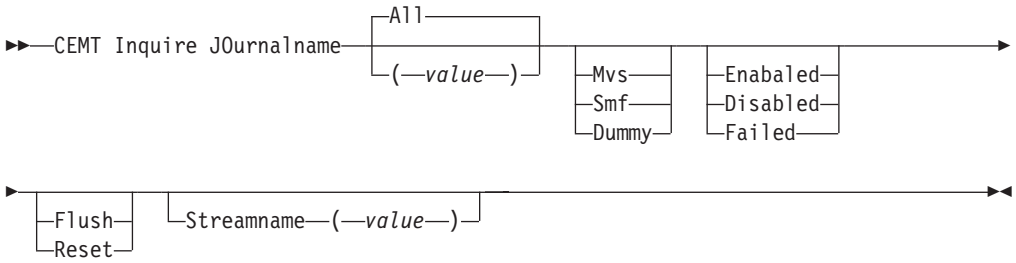
Note: Blank fields on the screen indicate that a value does not apply or is 'negative'; that is, it begins with 'No'.

If you place the cursor against a specific entry in the list, CICS displays an expanded format as shown in Figure 110.

```
IN JOUR
RESULT - OVERTYPE TO MODIFY
  Journalname(DFHLOG)
  Type(Smf)
  Status( Enabled )
  Action(             )
  Streamname(CICSHT##.CICSHTA1.DFHLOG)
```

Figure 110. The expanded display

CEMT INQUIRE JOURNALNAME



Displayed fields

Action (*value*)
specifies the action you want CICS to take for the specified journal name. The 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 the stream is processed using a batch utility.

Reset
The journal is disabled but can be reenabled by the SET JOURNALNAME RESET command.

Journalname (*value*)
indicates that this panel relates to a journal name inquiry and displays a 1– to 8–character journal name.

Status
displays the status of the journal. The possible values are:

Enabled
The journal is installed and available for use.

Disabled
The journal has been disabled by a CEMT, or EXEC CICS, SET JOURNALNAME(...) command. It cannot be used until it is re-enabled by the STATUS(ENABLED) or ACTION(RESET) options on a SET JOURNALNAME command.

Failed
The journal has experienced a log stream failure. It cannot be used until it is re-enabled by the STATUS(ENABLED) or ACTION(RESET) options on a a SET JOURNALNAME command, or until following the next CICS restart. The log stream should be deleted from the MVS system logger inventory before being used again.

Note: You can reset STATUS with ENABLED or DISABLED.

Streamname (*value*)
displays the MVS log stream name associated with the journal. CICS displays 8 blanks (X'40') if the journal is defined with a type of DUMMY or SMF.

Type
displays the type of log stream. The possible types are:

Mvs Records for journals defined with this journal model are written to an MVS log stream.

Smf Records for journals defined with this journal model are written to the MVS SMF data set.

Dummy
Records for journals defined with this journal model are not written to any log stream.

CEMT INQUIRE JVM

Identify JVMs in a CICS region and report their status.

Input

Enter `CEMT INQUIRE JVM` (or `CEMT I JVM`) to display all the JVMs in the CICS region. You cannot change the displayed values. You can also enter `CEMT INQUIRE JVM` with one of the values listed below, to display all the JVMs in the CICS region with a particular status. For example, you could enter `CEMT INQUIRE JVM UEXECKEY` to display all the JVMs that execute in user key, or `CEMT INQUIRE JVM PROFILE(DFHJVMPC)` to display all the JVMs that were created with the JVM profile DFHJVMPC.

If you are using the IBM SDK for z/OS, V1.4.2, to provide Java support, 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.

Sample screen

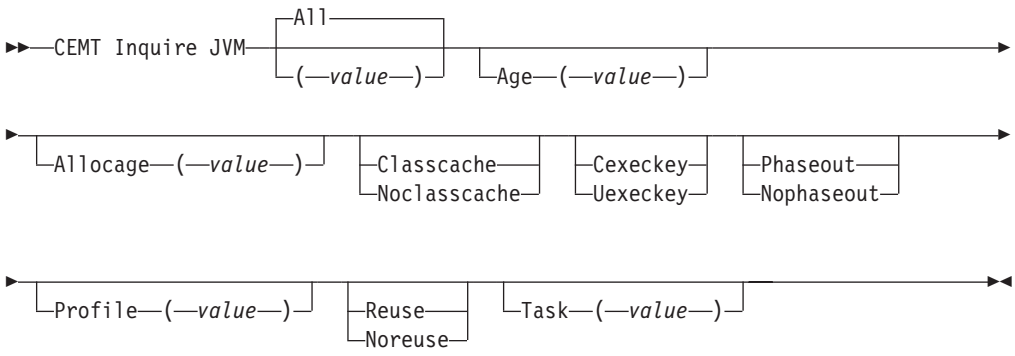
```
i jvm
STATUS:  RESULTS - OVERTYPE TO MODIFY
Jvm(0050463184) Age(0000001810) Allo(0000000000) Cla Reu
   Uex     Pro(DFHJVMPC) Tas(0000000)
Jvm(0050463485) Age(0000001804) Allo(0000000000)     Reu
   Uex     Pro(DFHJVMPR) Tas(0000000)
Jvm(0067240142) Age(0000001711) Allo(0000000000)     Reu
   Uex     Pro(DFHJVMPR) Tas(0000000)
Jvm(0084017268) Age(0000001668) Allo(0000000000) Cla Reu
   Uex     Pro(DFHJVMPC) Tas(0000000)
```

Figure 111. CEMT INQUIRE JVM screen

```
INQ JVM
RESULT - OVERTYPE TO MODIFY
  Jvm(0050463184)
  Age(00000001811)
  Allocage(0000000000)
  Classcachest(Classcache)
  Reusest(Reuse)
  Execkey(Uexeckey)
  Phasingoutst(Nophaseout)
  Profile(DFHJVMPC)
  Task(0000000)
```

Figure 112. The expanded display

CEMT INQUIRE JVM



Displayed fields

(*value***)**
is the JVM number, a unique value that identifies the JVM. The JVM number used in CICS is the same number that is used to identify the JVM in the z/OS UNIX environment, where it is known as the process id (PID) for the JVM.

Age(*value***)**
displays the number of seconds since the JVM was initialized.

Allocage(*value***)**
displays 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
shows whether the JVM uses the shared class cache. The 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
shows the execution key of the JVM. The values are:

Cexeckey
The JVM executes in CICS key.

Uexeckey
The JVM executes in user key.

Phasingoutst
shows whether the JVM is being phased out as a result of a CEMT SET JVMPOOL Phaseout, Purge or Forcepurge or CEMT PERFORM CLASSCACHE Phaseout, Purge or Forcepurge command (or the equivalent EXEC CICS commands). The 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(*value***)**
displays the 8–character name of the JVM profile used to initialize this JVM.

Reusest
shows whether or not this JVM can be reused. The values are:

Reuse The JVM profile for this JVM specified that it is continuous.

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

Task(*value***)**
displays the task to which the JVM is allocated, or zero if the JVM is not currently allocated to a task.

CEMT INQUIRE JVMPOOL

Retrieve information about the JVM pool, if any JVMs exist, in the CICS region.

Input

Press the Clear key to clear the screen then enter `CEMT INQUIRE JVMPOOL` (or `CEMT I JVMP`) to obtain information about the JVM pool. Note that there is no identifier on this command—a CICS region supports only one pool of JVMs.

Sample screen

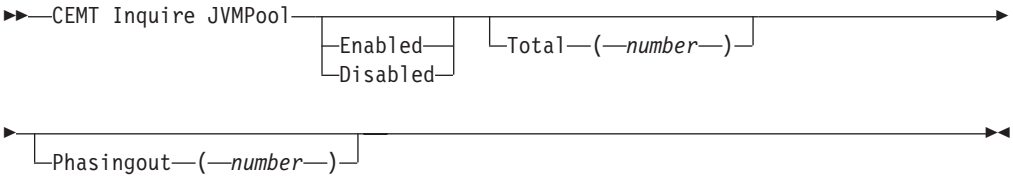
```
  i jvmp
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Jvm Ena Tot(0004) Phasi(0000)
```

Figure 113. CEMT INQUIRE JVMPOOL screen

```
INQ JVMPOOL
RESULT - OVERTYPE TO MODIFY
 Jvmpool
 Status( Enabled )
 Total(0004)
 Phasingout(0000)
 Terminate(              )
```

Figure 114. The expanded display

CEMT INQUIRE JVMPOOL



Displayed fields

Phasingout(*number***)**
displays 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.

Status(*value***)**
displays the overall status of the JVM pool. The possible values are:

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 (*number***)**
displays 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 in the PHASINGOUT count.

CEMT INQUIRE LIBRARY

Retrieve information about LIBRARY resources.

Function

Retrieve information about LIBRARY resources.

Description

CEMT INQUIRE LIBRARY returns information about the LIBRARY resources that are defined to your system.

You can reset the options either by typing the SET command or by overtyping at the appropriate place on the INQUIRE screen.

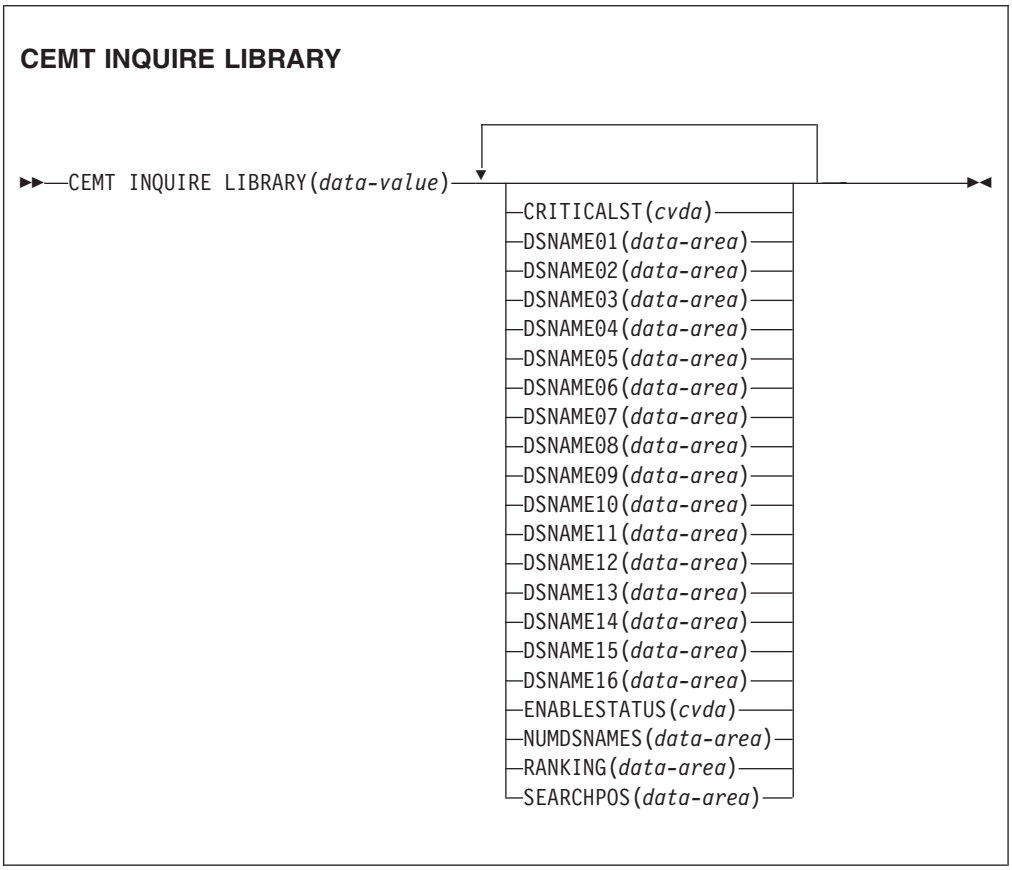
Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE LIBRARY` (the minimum abbreviation is `CEMT I LIB`). You get a display that lists the current status. LIBRARY resources are listed in ranking order, and by time of install within LIBRARYs of equal ranking.
- Type `CEMT INQUIRE LIBRARY` (`CEMT I LIB`) followed by as many of the other attributes as are necessary to limit the range of information that you require.

You can then tab to the highlighted or blank fields and overtype them with the required values.

(*data-value***)**
specifies the name of the LIBRARY resource. If you omit the LIBRARY name, the ALL option is assumed by default.



Displayed fields

CRITICALST(*cvda***)**
returns a CVDA value identifying whether the LIBRARY is critical to the start up of CICS. CVDA values are:

CRITICAL
The LIBRARY is critical to CICS startup. If the LIBRARY cannot be successfully installed during CICS startup for any reason, a 'GO or CANCEL' message is issued to allow the operator to decide whether to override the criticality and allow CICS to start. If the operator selects

'GO', the LIBRARY is installed in a 'disabled' status, unless install is not possible at all; for example, due to a short-on-storage condition.

Note: If the reply is to continue with the startup, the LIBRARY will not be recatalogued as NONCRITICAL, so the critical status should be explicitly set to NONCRITICAL if it is decided that the LIBRARY should not be regarded as CRITICAL in future.

NONCRITICAL
The LIBRARY is not critical to CICS startup. If the LIBRARY cannot be successfully installed during CICS startup, the LIBRARY is left in an installed but disabled state and a warning message will be issued, but CICS startup continues.

LIBRARY(*data-value***)**
Specifies the 8-character name of the LIBRARY about which you are inquiring. If you omit the program name, the ALL option is assumed by default.

DSNAME01-16(*data-area***)**
Return the names of data sets in the LIBRARY concatenation. If this library is dynamically-defined, these are the data sets specified on the LIBRARY definition, all but one of which can be blank. If this DFHRPL is statically-defined, these are the first 16 data sets in the DFHRPL concatenation, or as many data sets as are specified up to 16, with the remaining DSNAMExx fields being blank.

If there are more than 16 data sets in the DFHRPL concatenation, you can use either a CICSPlex SM LIBDSN view to list all of the data sets or the DSNAMELIST option on the **EXEC CICS INQUIRE LIBRARY** command.

ENABLESTATUS(*cvda***)**
Returns a CVDA value identifying whether the LIBRARY is included in the overall LIBRARY search order. CVDA values are:

DISABLED
The LIBRARY is disabled, and is not currently included in the LIBRARY search order. The data sets in this LIBRARY concatenation are not searched for program artifacts.

DISABLING
A request to disable the LIBRARY has been received, but is still being processed.

ENABLED
The LIBRARY is enabled, and is currently included in the LIBRARY search order. The data sets in this LIBRARY concatenation will be searched for program artifacts to load.

ENABLING
A request to enable the LIBRARY has been received, but is still being processed.

DISCARDING
A request to discard the LIBRARY from the CICS system has been received, but is still being processed.

NUMDSNAMES(*data-area***)**
Specifies the number of data sets in the LIBRARY concatenation. For a dynamically defined LIBRARY, this will be the number of non-blank DSNAMExx

values, and cannot be a value larger than 16. For the statically defined DFHRPL, this is the number of data sets in the concatenation, and can be a value larger than 16.

RANKING(*data-area***)**
Specifies where this LIBRARY should appear in the overall LIBRARY search order relative to other LIBRARY concatenations. A lower number indicates that this LIBRARY is searched for programs to load before other LIBRARY resources with higher ranking numbers. However LIBRARYs of equal RANKING will appear in the search order in the order in which they were installed or created in the local CICS system, with a LIBRARY that was installed earlier appearing before one that was installed later.

SEARCHPOS(*data-area***)**
Returns the current absolute position of this LIBRARY in the overall LIBRARY search order. The first LIBRARY in the search order has a SEARCHPOS of 1, the next LIBRARY has a SEARCHPOS of 2, and so on. The SEARCHPOS is not the same as the RANKING, although its value is determined by the relative ranking values of the various LIBRARY resources in the system.

CEMT INQUIRE LINE

Retrieve information about lines connected to terminals. This request is valid only for non-VTAM terminals.

Input

Press the Clear key to clear the screen. There are two ways to start this transaction:

- Type `CEMT INQUIRE LINE(`*value*`)` (the minimum abbreviation is `CEMT I LIN(`*value*`)`). The current status of the named line is displayed.
- Type `CEMT INQUIRE LINE(termid)` (`CEMT I LIN(termid)`) followed by as many attributes as appropriate to limit the range of information that you require.

You can specify a list of identifiers (see “List of resource identifiers” on page 128), but you cannot use the symbols * and + to specify a family of identifiers. You can then tab to the highlighted or blank fields and overtype them with the required values.

Sample screen

```
INQ LINE(TTT1)
STATUS:  RESULTS
 Lin(TTT1)         Neg( 00080 ) Ins
```

Figure 115. CEMT INQUIRE LINE screen

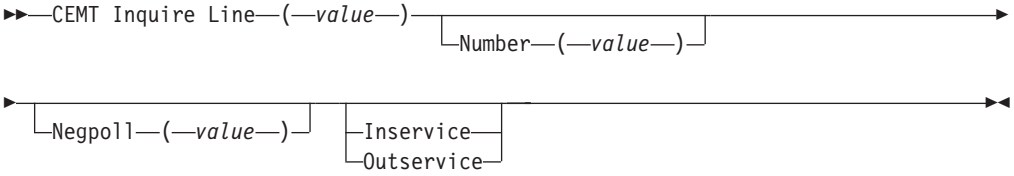
Note: The screen has blank fields where a value does not apply or is negative; that is, the value begins with “No”. To modify such fields, tab to locate each one (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. It can be easier to use the expanded format to set such a value (setting a value to its negative is easier because you overtype a displayed field with its negative, which results in a blank field).

If you place the cursor against a specific entry in the list, CICS displays an expanded format as shown in Figure 116.

```
INQ LINE(TTT1)
RESULT
  Line(TTT1)
  Number()
  Negpoll( 00080 )
  Servstatus( Inservice )
  Purgetype(            )
```

Figure 116. The expanded display

CEMT INQUIRE LINE



Displayed fields

Line (*value*)
indicates that this panel relates to a LINE inquiry and displays the identifier of a terminal on the line. This request is valid only for non-VTAM terminals.

Number (*value*)
displays the relative line number of a pooled line. For information about pooled lines, see the *CICS Resource Definition Guide*.

Servstatus
The possible values are:
Inservice
The line is available for use. Changing the status of the line does not affect the equivalent status of the terminals on it.
Outservice
The line is not available for use. Changing the status of the line does not affect the equivalent status of the terminals on it.

CEMT INQUIRE MODENAME

Retrieve information about connection sessions.

Description

INQUIRE MODENAME returns information about the named group of sessions that has been defined for a connection to a remote system or to another CICS region.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE MODENAME` (the minimum abbreviation is `CEMT I MOD`). You get a display that lists the current status.

- Type CEMT INQUIRE MODENAME (CEMT I MOD) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter cemt i mod acq, the resulting display will show you the details of all modenames that are acquired.

Mode names need not be unique. You cannot specify a list of names, but you can use the symbols * and + to specify a family of mode names.

A mode name consisting of eight blanks is valid. To specify this name, enclose the string of blanks within single quotation marks:

```
MODENAME('   ')
```

You can then tab to the highlighted or blank fields and overtype them with the required values.

(*value***)**
specifies one or more names (1-8 characters) of a group of sessions.

ALl
is the default.

Sample screen

```
I MODE
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Mod(SNASVCMG) Con(C   ) Max(002) Ava( 000 ) Act(000)
 Mod(MODE1   ) Con(C   ) Max(001) Ava( 000 ) Act(000)
```

Figure 117. CEMT INQUIRE MODENAME screen

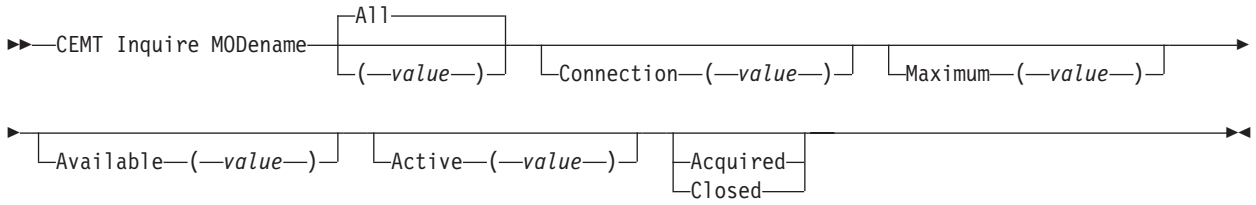
Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against a specific entry in the list, CICS displays an expanded format as shown in Figure 118.

```
I MODE
RESULT - OVERTYPE TO MODIFY
  Modename(MODE1)
  Connection( C    )
  Maximum(001)
  Available( 000 )
    Active(000)
    Acqstatus(             )
```

Figure 118. The expanded display

CEMT INQUIRE MODENAME



Displayed fields

Acqstatus (*input only field***)**
specifies the acquisition status of sessions. The values are:

Acquired
CICS should acquire (or "bind") all the negotiated contention-winner sessions within the group.

Closed
CICS releases all sessions within the group. The sessions are quiesced and unbound. No further sessions can be acquired by another CICS system. No further activity is possible on the sessions within the group until the group is reacquired by the same CICS system in which it has been closed.

Active (*value***)**
displays the number of sessions within the group (or "modename") that are currently in use ("bound").

Available (*value***)**
displays the currently negotiated number of sessions within the group that can be allocated for use at one time.

Note: You can reset this value by overtyping it with a different value.

Connection (*value***)**
displays the 4-character identifier of the remote system with which this group of IRC or ISC sessions is connected.

Maximum (*value***)**
displays the maximum number of sessions that are supported at one time within this group of sessions, in the range 0–999.

Modename (*value***)**
indicates that this panel relates to a MODENAME inquiry and displays one or more names (1-8 characters) of a group of sessions.

CEMT INQUIRE MONITOR

Retrieve information about the status of CICS monitoring.

Description

INQUIRE MONITOR tells you whether monitoring is switched on or off, which classes of monitoring data are to be collected, and the current settings of the monitoring options.

Input

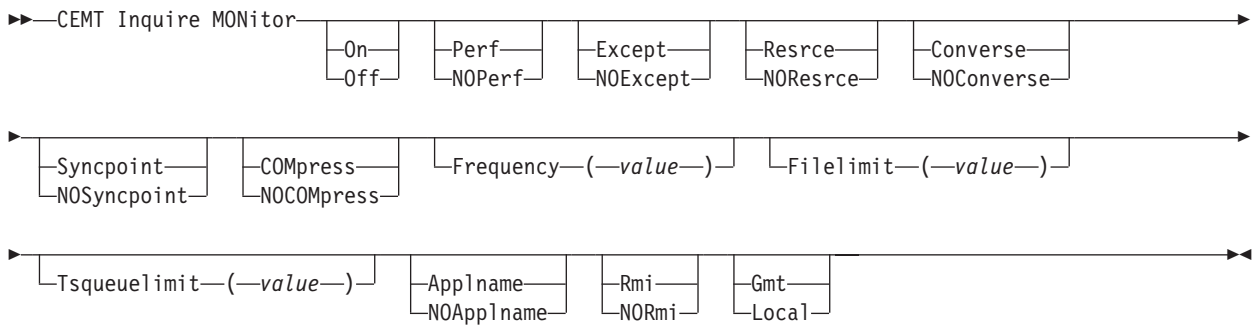
Press the Clear key to clear the screen, and type CEMT INQUIRE MONITOR (the minimum abbreviation is CEMT I MON). You get a display that lists the current status. You can then tab to the highlighted or blank fields and overtype them with the required values.

Sample screen

```
INQ MON
RESULT - OVERTYPE TO MODIFY
  Status( On )
  Perfclass( Perf )
  Exceptclass( Except )
  Resrceclass( Resrce )
  Conversest( Noconverse )
  Syncpointst( Nosyncpoint )
  Compressst( Compress )
  Frequency( 000000 )
  Filelimit(16)
  Tsqueuelimit(08)
  Applnamest(Applname)
  Rmist(Normi)
  Time(Local)
```

Figure 119. CEMT INQUIRE MONITOR screen

CEMT INQUIRE MONITOR



Displayed fields

Applnamest
displays whether CICS application naming support is enabled. The values are:

Applname
CICS application naming support is enabled.

Noapplname
CICS application naming support is not enabled.

COMpressst
displays whether data compression is performed for monitoring records. The values are:

Compress
Data compression is performed.

Nocompress
Data compression is not performed.

Note: You can reset this value by overtyping it with a different value.

Conversest
displays whether conversational tasks are to have separate performance class records produced for each pair of terminal control I/O requests (Converse or Send/Receive pair). The values are:

Converse
Separate performance class records are produced.

Noconverse
Separate performance class records are not produced.

Note: You can reset this value by overtyping it with a different value.

Exceptclass
displays whether the exception class of monitoring data is being collected. The values are:

Except
Exception-class monitoring data is being collected.

Noexcept
Exception-class monitoring data is not being collected.

Note: You can reset this value by overtyping it with a different value.

Filelimit
displays the maximum number of files for which transaction resource class monitoring data is being collected.

Frequency (*value***)**
displays the interval for which monitoring automatically produces a transaction performance class record for any long-running transaction that has been in the system for a greater time than the interval. FREQUENCY values can be either 000000 (meaning that FREQUENCY monitoring is inactive), or in the range 000100–240000.

Note: You can reset this value by overtyping it with a different value.

MONitor
indicates that this panel relates to a MONITOR inquiry.

Perfclass
displays whether the performance class of monitoring data is being collected. The values are:

Perf Performance-class monitoring data is being collected.

Noperf
Performance-class monitoring data is not being collected.

Note: You can reset this value by overtyping it with a different value.

Resrceclass
displays whether the transaction resource class of monitoring data is being collected. The values are:

Resrce
Transaction resource class monitoring data is being collected.

Noresrce
Transaction resource class monitoring data is not being collected.

Note: You can reset this value by overtyping it with a different value.

Rmist
displays whether additional performance class monitoring data is to be collected for the resource managers used by your transactions. The values are:

Rmi Additional performance class monitoring data is being collected for the resource managers used by your transactions.

Normi Performance class monitoring data is not being collected for the resource managers used by your transactions.

Status
displays whether monitoring is active. The values are:

On Monitoring data is accumulated continuously, and is collected (written out to an SMF data set) for each of the classes of monitoring data that is active.

Off No monitoring data is accumulated or collected, irrespective of the settings of the monitoring data classes.

Note: You can reset this value by overtyping it with a different value.

Syncpointst
displays whether a transaction performance class record is produced when a transaction takes an explicit or implicit syncpoint (unit-of-work), except when the syncpoint is part of task termination or a syncpoint rollback. The values are:

Syncpoint
A transaction performance class record will be produced.

Nosyncpoint
A transaction performance class record will not be produced.

Note: You can reset this value by overtyping it with a different value.

Time
displays whether the time stamp fields in the performance class monitoring data are returned in GMT or local time to an application using the **COLLECT STATISTICS** command. The values are:

Gmt The time stamp fields in the performance class data are Greenwich mean time (GMT).

Local The time stamp fields in the performance class data are local time.

TSqueuelimit
displays the maximum number of temporary storage queues for which transaction resource class monitoring data is being collected.

CEMT INQUIRE NETNAME

Retrieve information about a network.

Description

If you are familiar with network names, you might prefer to use this command as an alternative to CEMT INQUIRE TERMINAL.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE NETNAME` (the minimum abbreviation is `CEMT I N`). You get a display that lists the current status.
- Type `CEMT INQUIRE NETNAME` (`CEMT I N`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So,

for example, if you enter `cemt i n ins ati`, the resulting display indicates the details of netnames that are in service and available for use.

You can then use the tab key to the highlighted or blank fields and overtype them with the required values.

(*value***)**
specifies an 8-character network name.

ALl
is the default.

Sample screen

```
IN NETNAME
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Net(IGKS205 ) Tra(CEMT) Pri( 000 ) Pag Ins Ati Tti
         Ter(TC03) Acq     Tas(0000198)           Nqn(GBIBMIYA.IYCQTC03)
```

Figure 120. CEMT INQUIRE NETNAME screen

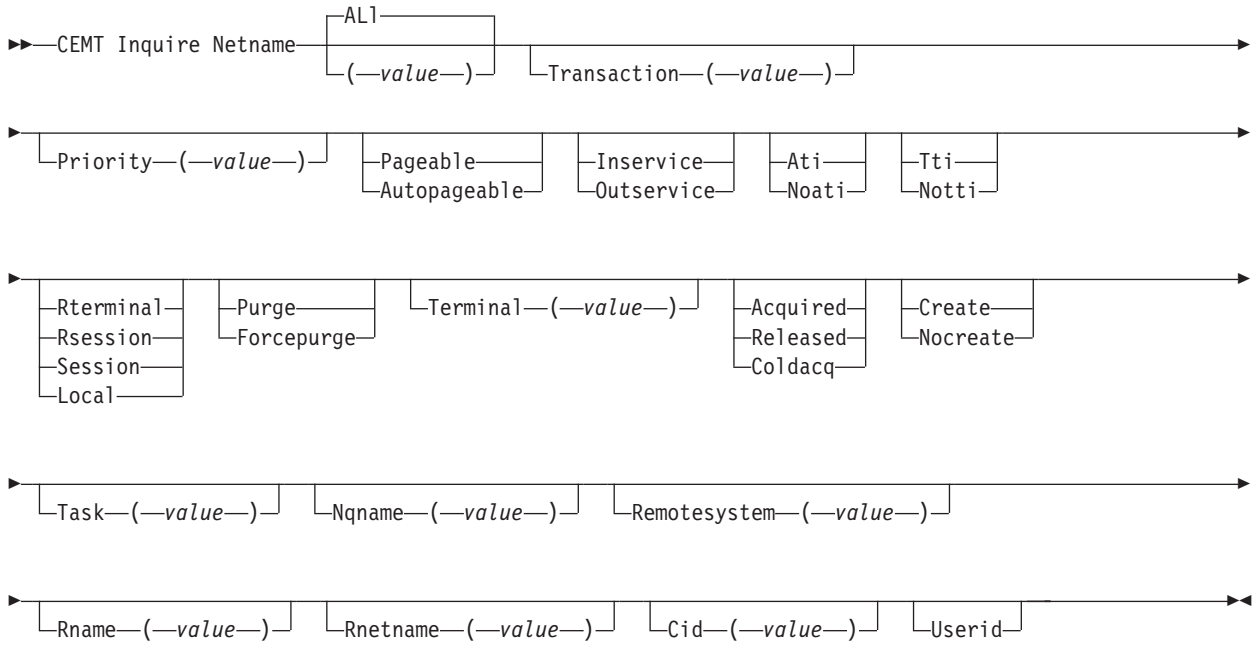
Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by pressing tab (they are shown in the same sequence as in the expanded format), and overtype with input valid for that field. You can also use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative, and this prevents the result from being displayed).

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 121.

```
IN NET
RESULT - OVERTYPE TO MODIFY
  Netname(IYCQTC03)
  Transaction(CEMT)
  Priority( 000 )
  Pagestatus( Pageable )
  Servstatus( Inservice )
  Atistatus( Ati )
  Ttistatus( Tti )
  Nature(Local)
  Purgetype(              )
  Terminal(944D)
  Termstatus( Acquired )
  Createsess( Nocreate )
  Task(00033)
  Remotesystem()
  Nqname(GBIBMIYA.IYCQTC03)
  Rname()
  Rnetname()
  Cid()
  Userid()
```

Figure 121. The expanded display of an individual entry

CEMT INQUIRE NETNAME



Displayed fields

AIDCOUNT (*data-area***)**
Returns a fullword binary field giving the number of Automatic Initiate Descriptors (AIDs) queued for the specified terminal. If there are no AIDs, then an AIDCOUNT value of 0 is returned. The count represents the number of AIDs on the CICS system that issued the command.

Atistatus
displays whether the terminal is available for use 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. The values are:
Ati The terminal is available for use.
Noati The terminal is not available for use.

Note:

1. You can reset this value by overtyping it with a different value.
2. A terminal cannot be defined with both NOATI and NOTTI.

Cid (*value***)**
displays an 8-character correlation-identifier that is set only for sessions, and only after a session has been acquired. (The value relates to the previously acquired session if there was one, if the session is not acquired). CID is set as follows:

- For LU6.2 sessions, it is an 8-character hexadecimal token which is common to the two end sessions that are connected together.
- For MRO sessions, it is set to the termid of the session on the system to which this session is connected.

- For LU6.1 sessions to IMS, it is set to the value of NETNAMEQ as defined or negotiated.
- For other LU6.1 sessions, it is set to the value supplied by the other end of the 6.1 session.

Using CID, you can relate the two parts of an LU6.2, MRO, or LU61 conversation together:

- For LU6.2, you can inquire on the same CID.
- For MRO, you can locate the session with the terminal name.
- For LU6.1 to IMS, you can inquire on the netname.
- For other LU6.1, you can use the name to inquire on the connected system.

Createsess (*vtam only*)
displays whether the terminal can be acquired automatically by ATI transactions. This cannot be specified for IRC sessions. The values are:

Create
If the terminal is not in session, CICS acquires it if it is needed to satisfy an ATI request.

Nocreate
If the terminal is not in session, CICS does not acquire it to satisfy an ATI request. A session must be started by, for example, a logon request or a CEMT SET TERMINAL ACQUIRED command before the ATI request can be satisfied.

If NOCREATE is set for an LU6.1 ISC session and there are no allocatable sessions left, the connection is placed OUTSERVICE.

Note: You can reset this value by overtyping it with a different value.

Nature
displays the nature of the terminal—that is, whether it is a physical device or a session, and whether it is local or remote. The values are:

Rterminal
The terminal is owned by another CICS region.

Rsession
The terminal is a session of a connection that is owned by another CICS region.

Session
The terminal is an ISC or IRC session.

Local The terminal is directly attached to this system and is a physical device or a function of a cluster controller.

Netname (*value*)
indicates that this panel relates to a NETNAME inquiry and displays an 8-character network name.

- For a physical terminal, the Netname is the name by which this terminal is known to VTAM.
- For ISC sessions, the Netname is the name by which the session (or group of sessions) is known to VTAM.
- For IRC sessions, the Netname is the name used by the connected region to logon to the interregion communication program (DFHIRP).
- For EXCI sessions, the Netname is **DFHGEN** for generic sessions and, for specific sessions, it is the value of the user_name parameter used in the EXCI INITIALISE_USER call.

When an inquiry is made on DFHGEN, the name is an 8-character field with a leading and trailing blank (' DFHGEN '). The name must be enclosed in quotes and include the blanks, so you enter the command on the screen as follows:

```
CEMT INQ NETNAME(' DFHGEN ')
```

- For remote devices, the Netname is the name by which the device is known to the VTAM in the terminal-owning region.

Note: If the netname is a VTAM LU alias, it is different from the netname component of the NQNAME, which always contains the real netname.

Nqname
displays the VTAM 17-character network-qualified name.

Remote terminals do not have an NQNAME value.

If the status (Termstatus) is RELEASED, NQNAME displays the netname.

Pagestatus
displays whether pages after the first in a series are written to the terminal on request from the operator or automatically. The values are:

Pageable
Pages are written on request.

Autopageable
Pages are written automatically.

Note: You can reset this value by overtyping it with a different value.

Priority (*value*)
displays a 3-character string indicating the priority of this terminal relative to other terminals. The priority of a task is the sum of the transaction priority, the terminal priority, and the operator priority. Priority has no meaning for terminals that are ISC sessions being used as alternative facilities. The value is in the range 0–255, where 255 is the highest priority.

Note: You can reset this value by overtyping it with a different value.

Purgetype (*input only field*)
specifies whether the transactions running with the named terminal are to be purged. The values are:

Purge
Transactions are to be terminated only if system and data integrity can be maintained. A transaction is to be purged if its definition specifies SPURGE=NO.

Forcepurge
Transactions are to be purged immediately. This can lead to unpredictable results and should be used only in exceptional circumstances.

Remotesystem (*value*)
displays 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 terminal-owning region (TOR), or an indirect connection that provides the netname of the TOR.

Otherwise this field is blank.

Rname (*value*)
displays the 4-character name by which this terminal is known in the system in

which it is defined (the TOR). RNAME applies only to terminals defined as remote; for others, blanks are displayed.

Rnetname (*value*)
displays the 8-character netname of the owning TOR, if the subject of the inquiry is a remote terminal. If this field is blank **and** the terminal is remote, the system named in the REMOTESYSTEM field has not been installed, and no value was specified for the REMOTESYSNET option when the terminal was defined.

Servstatus
displays whether the terminal is available for use. The values are:

Inservice
The terminal is available for use. For VTAM, INSERVICE means that the terminal can be ACQUIRED. For IRC sessions, INSERVICE means that the connection to the MRO partner is INSERVICE.

Outservice
The terminal is not available for use. Setting a terminal OUTSERVICE means that the terminal can no longer be used by transactions. If PURGE or FORCEPURGE is also specified, any transaction using the terminal is terminated abnormally. If PURGE or FORCEPURGE is not specified, the transaction is allowed to terminate normally, but no further transactions are allowed to use the terminal. For VTAM, setting a terminal OUTSERVICE also causes it to be released and the operator to be signed off, either immediately or when the current transaction has terminated.

For IRC sessions, OUTSERVICE means that the connection to the MRO partner is OUTSERVICE.

In an LU6.1 ISC session, the connection is set OUTSERVICE if there are no allocatable sessions left.

Note: For physical terminals and LU6.1 sessions, you can reset this value by overtyping it with a different value.

Task (*value*)
displays the task number of the transaction that is running on this terminal.

Terminal (*value*)
displays a 4-character terminal identifier (1-4 characters) as specified in an installed terminal definition.

Termstatus (*vtam only*)
displays whether CICS is in session with the logical unit represented by this terminal. The values are:

Acquired
CICS is in session with the logical unit represented by the terminal.

Released
CICS is not in session with the logical unit represented by the terminal. If you set this option to RELEASED, a session is terminated immediately if you also specify the PURGE option, otherwise the session is terminated when the current active transaction finishes.

Note: For physical terminals and LU6.1 sessions, you can reset this value by overtyping it with a different value or with:

Coldacq
This is a special form of ACQUIRED, where no resynchronization is required. If the previous session abended, the use of

COLDACQ overrides CICS integrity control. This might lead to integrity problems. Also check the CSMT log for an activity keypoint after the restart of a session following a CICS failure. If there is no activity keypoint, issue COLDACQ again after the next emergency restart.

Transaction (*value***)**
displays a 4-character string indicating the name of the transaction currently being processed with this terminal as its principal facility or as a secondary facility.

Ttistatus
displays whether the terminal can be used by the transactions that are initiated from this terminal. The values are:
Tti This terminal can be used by transactions.
Notti This terminal cannot be used by transactions.

Note:

1. You can reset this value by overtyping it with a different value.
2. A terminal cannot be defined with both NOATI and NOTTI.

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 userid—as specified in the DFLTUSER system initialization parameter—is returned.

CEMT INQUIRE PARTNER

Retrieve information about the partner table.

Description

INQUIRE PARTNER returns information from the partner table (side information table) installed in your system. There is no SET PARTNER function. However the INQUIRE PARTNER screen does allow you to type a `D` in the left-hand margin by the partnername to be discarded (the CEMT DISCARD PARTNER function).

For information on partners, see the *CICS Resource Definition Guide*.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE PARTNER` (the minimum abbreviation is `CEMT I PA`). You get a display that lists the current status.
- Type `CEMT INQUIRE PARTNER` (`CEMT I PA`) followed by as many of the other attributes as are necessary to limit the range of information that you require.

(*value***)**
specifies the 8-character partner name as specified in the installed partner definition.

All
is the default.

Sample screen

```
I PART
STATUS:  RESULTS
  Par(IST1ISC2)               Netn(IYMVM2  ) Pro(DFHCICSA)
     Ltp(04) Tpn(IST1                      )
  Par(IST2ISC2)               Netn(IYMVM2  ) Pro(DFHCICSA)
    Ltp(04) Tpn(IST2                      )
```

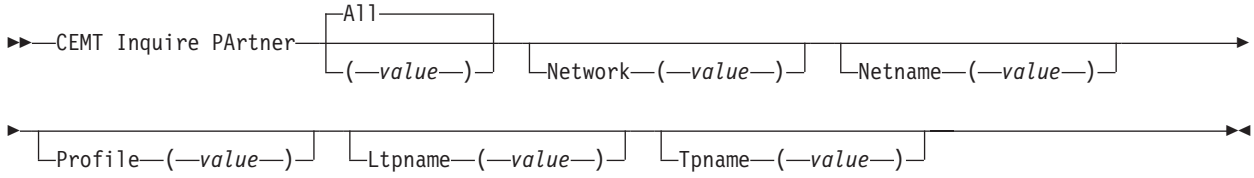
Figure 122. CEMT INQUIRE PARTNER screen

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 123.

```
I PART
RESULT
  Partner(IST1ISC2)
  Network()
  Netname(IYMVM2)
  Profile(DFHCICSA)
  Ltpname(04)
  Tpname(IST1)
```

Figure 123. The expanded display of an individual entry

CEMT INQUIRE PARTNER



Displayed fields

Ltpname(*value*)
displays the length of the name of the remote transaction program (TPNAME), in bytes.

Netname(*value*)
displays the 8-character name of the node at which the partner is located.

Network(*value*)
displays the 8-character name of the network in which the partner is located.

Partner(*value*)
indicates that this panel relates to a PARTNER inquiry and displays the 8-character partner name as specified in the installed partner definition.

Profile(*value*)
displays the 8-character name of the CICS profile associated with the partner.

Tpname(*value*)
displays the 64-character name of the remote transaction program.

Note: Nondisplayable characters appear as periods. You can use the PF2 key on the expanded panel to see the hexadecimal value.

CEMT INQUIRE PIPELINE

Retrieve information about an installed pipeline.

Description

Use the **INQUIRE PIPELINE** command to retrieve information about an installed pipeline.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE PIPELINE` (the minimum abbreviation is `CEMT I PI`). You get a screen that lists the current status.
- Type `CEMT INQUIRE PIPELINE` (`CEMT I PI`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i pi en`, the resulting display will show you the details of only those process-types that are enabled.

To change various attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field.
- Use the **CEMT SET PIPELINE** command.

ALl
is the default. Information about all pipelines is displayed, unless you specify a selection of pipelines to be queried.

(value)
is the name (1-8 characters) of an installed PIPELINE definition.

Displayed fields

Ciddomain*(value)*
Displays the name of the domain that is used to generate MIME content-ID values that identify binary attachments.

Configfile*(value)*
Displays the name of the pipeline configuration file associated with the PIPELINE resource. The name can be up to 255 characters long.

Enablestatus
Displays 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.

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.

Mode*(value)*
Displays the operating mode of the pipeline.

PROVIDER
CICS is using the pipeline as a service provider of Web services.

REQUESTER
CICS is using the pipeline as a service requester of Web services.

UNKNOWN
The operating mode of the pipeline cannot be determined.

Mtomnoxopst*(value)*
Displays the status of the pipeline for sending outbound messages in MIME format when binary attachments are not present.

MTOMNOXOP
Outbound messages are sent in MIME format, even when there are no binary attachments present.

NOMTOMNOXOP
Outbound messages are only sent in MIME format when there are binary attachments present.

Mtomst*(value)*
Displays the status of the MTOM handler in the pipeline.

MTOM
The MTOM handler is enabled in the pipeline.

NOMTOM
The MTOM handler is not enabled in the pipeline.

Respwait *(number)*
Displays the number of seconds that an application program waits for an optional message from a remote Web service. If no value is displayed, the default timeout value of the transport protocol is being used.

- The default timeout value for HTTP is 10 seconds.
- The default timeout value for WebSphere MQ is 60 seconds.

Sendmtomst*(value)*
Displays the status of the pipeline for sending outbound messages in MIME format.

NOSENDMTOM
Outbound messages are never sent in MIME format.

SAMESENDMTOM
Outbound messages are only sent in MIME format when the inbound message is in MIME format.

SENDMTOM
Outbound messages are always sent in MIME format.

Shelf*(value)*
Displays the name of the *shelf directory*. The name can be up to 255 characters long.

SOAPlevel*(value)*
Displays the level of SOAP that is supported in the pipeline. The SOAP level can be 1.1 or 1.2. If the pipeline is not being used for SOAP messages, a value of NOTSOAP is displayed.

Wsdir*(value)*
Displays the name of the *Web service binding directory* (also known as the *pickup directory*). The name can be up to 255 characters long.

Xopdirectst*(value)*
Displays the status of the pipeline for handling XOP documents and binary attachments in direct or compatibility mode.

XOPDIRECT
The pipeline is processing XOP documents and binary attachments in direct mode.

NOXOPDIRECT
The pipeline is processing XOP documents and binary attachments in compatibility mode.

Xopsupportst*(value)*
Displays the status of the application handler for processing XOP documents and binary attachments directly.

XOPSUPPORT
The application handler supports the direct handling of XOP documents and binary attachments.

NOXOPSUPPORT
The application handler does not support the direct handling of XOP documents and binary attachments.

CEMT INQUIRE PROCESSTYPE

Retrieve information about a CICS business transaction services process-type.

Description

INQUIRE PROCESSTYPE returns information about the BTS PROCESSTYPE definitions installed on this CICS region. In particular, it shows the current state of audit logging for each displayed process-type.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE PROCESSTYPE` (the minimum abbreviation is `CEMT I PROC`). You get a screen that lists the current status.
- Type `CEMT INQUIRE PROCESSTYPE` (`CEMT I PROC`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i proc en`, the resulting display will show you the details of only those process-types that are enabled.

To change various attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field.
- Use the CEMT SET PROCESSTYPE command.

ALl
is the default. Information about all process-types is displayed, unless you specify a selection of process-types to be queried.

(value)
is the name (1-8 characters) of one or more PROCESSTYPE definitions installed in the process-type table (PTT).

Sample screen

```
I PROC
STATUS:   RESULTS - OVERTYPE TO MODIFY
 Pro(PROCESSTYPE12  ) Fil(FILE12  ) Aud(ADTLOG12) Pro Ena
 Pro(PROCESSTYPE13  ) Fil(FILE13  ) Aud(ADTLOG12) Off Ena
 Pro(PTYPE2B        ) Fil(FILE2B  ) Aud(DFHJ2B  ) Ful Ena
 Pro(PTYPE39        ) Fil(FILE39  ) Aud(DFHJ39  ) Off Ena
 Pro(SALESTYPE1     ) Fil(SALESF1 ) Aud(PLOG51  ) Off Dis
 Pro(SALESTYPE4     ) Fil(SALESF4 ) Aud(PLOG51  ) Act Ena
 Pro(SALESTYPE6     ) Fil(SALESF6 ) Aud(PLOG51  ) Off Ena
```

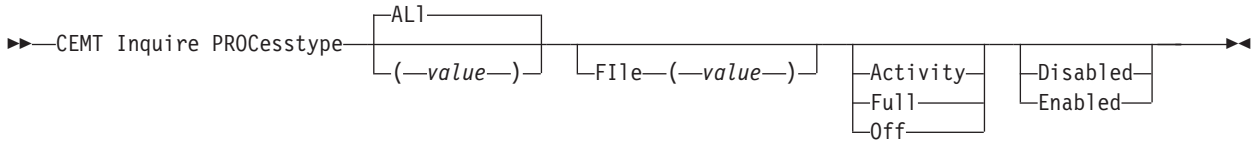
Figure 124. CEMT INQUIRE PROCESSTYPE screen

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 125.

```
 I PROC
 STATUS:   RESULTS - OVERTYPE TO MODIFY
   Processtype(SALESTYPE4     )
   File(SALESF4 )
   Enablestatus( Enabled )
   Auditlog(PLOG51  )
   Auditlevel(Activity )
```

Figure 125. The expanded display of an individual entry

CEMT INQUIRE PROCESSTYPE



Displayed fields

Auditlevel
displays the level of audit logging currently active for processes of this type. The 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*.

Auditlog*(value)*
displays the 8-character name of the CICS journal used as the audit log for processes of this type.

Enablestatus
displays whether new processes of this type can be created. The 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.

File*(value)*
displays the 8-character name of the CICS repository file on which the process and activity records for processes of this type are stored.

PROCesstype*(value)*
indicates that this panel relates to a PROCESSTYPE inquiry and displays the 8-character name of a process-type.

CEMT INQUIRE PROFILE

Retrieve information about a transaction profile.

Description

INQUIRE PROFILE returns information about the profiles installed in your system. There is no SET PROFILE function. However, the INQUIRE PROFILE screen does allow you to type a “D”. in the left-hand margin by the name of the profile to be discarded (the DISCARD PROFILE function).

Input

Press the Clear key and type `CEMT INQUIRE PROFILE` (the minimum abbreviation is `CEMT I PROF`).

You will get a display screen, a sample of which is shown below.

(*value***)**
is the name of a specific profile.

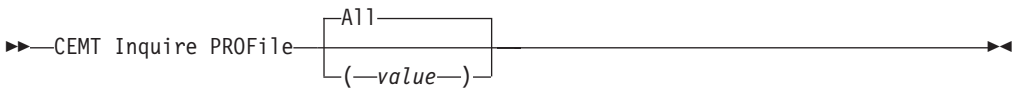
All
is the default.

Sample screen

```
IN PROFILE
STATUS:  RESULTS
 Prof(DFHCICSA)
 Prof(DFHCICSE)
 Prof(DFHCICSF)
 Prof(DFHCICSR)
 Prof(DFHCICSS)
 Prof(DFHCICST)
 Prof(DFHCICSV)
```

Figure 126. CEMT INQUIRE PROFILE screen

CEMT INQUIRE PROFILE



Displayed fields

PROFile(*value***)**
indicates that this panel relates to a PROFILE inquiry and displays the 8-character name of a profile installed in your system.

CEMT INQUIRE PROGRAM

Retrieve information about programs, maps, or partition sets.

Description

INQUIRE PROGRAM returns information about the programs, map sets, and partition sets that are defined to your system.

Note: Java programs that run in a JVM return a subset of values.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE PROGRAM` (the minimum abbreviation is `CEMT I PROG`). You get a display that lists the current status.
- Type `CEMT INQUIRE PROGRAM` (`CEMT I PROG`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i prog c e`, the resulting display will show you the details of only those programs that are written in C language and are enabled.

You can then tab to the highlighted or blank fields and overtype them with the required values.

(*value***)**
is a program identifier (1–8 characters), which is the name of a specific program entry in the table of installed program definitions. If you omit the program name, the ALL option is assumed by default.

You cannot specify the CLASS option and a program name.

ALl
is the default.

CLass(*value***)**
is the 2-character suffix of a program list table (PLT). Use this option to inquire on those programs defined in the specified PLT. Note that, unlike a shutdown PLT, it is not mandatory to define a program definition for a startup PLT. However, if you want to use the CLASS option, you must define and install a program resource definition for the PLT specified by “clasid”.

If the clasid specified is the suffix of a PLT that does not have a program definition, the INQUIRE PROGRAM CLASS(clasid) command returns a ‘CLASS NOT FOUND’ message.

Sample screen

```
 IN PROG
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Prog(CNCLTASK) Len(0000000000) Cob Pro Ena Pri    Ced
     Res(000) Use(0000000) Bel Uex Ful
  Prog(DFH$AALL) Len(0000002936) Ass Pro Ena Pri Ced
     Res(000) Use(0000000) Bel Uex Ful
  Prog(DFH$ABRW) Len(0000002520) Ass Pro Ena Pri     Ced
   Res(000) Use(0000000008) Bel Uex Ful Qua           Nat
  Prog(DFH$ABRW) Len(0000002520) Ass Pro Ena Pri Ced
     Res(000) Use(0000001) Bel Uex Ful
  Prog(DFH$ACOM) Len(0000001128) Ass Pro Ena Pri Ced
     Res(000) Use(0000000) Bel Uex Ful
  Prog(DFH$AGA ) Len(0000000520) Map Ena Pri
     Res(000) Use(0000002)
  Prog(DFH$AGB ) Len(0000000448) Map Ena Pri
     Res(000) Use(0000000)
  Prog(DFH$AGC ) Len(0000000512) Map Ena Pri
     Res(000) Use(0000000)
  Prog(DFH$AGD ) Len(0000000576) Map Ena Pri
     Res(000) Use(0000000)
+ Prog(DFH$AGK ) Len(0000000392) Map Ena Pri
     Res(000) Use(0000000)
```

Figure 127. CEMT INQUIRE PROGRAM screen

Note: There are blank fields on the screen where a value does not apply or is ‘negative’; that is, the value begins with ‘No’. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 128 on page 256 and Figure 129 on page 256.

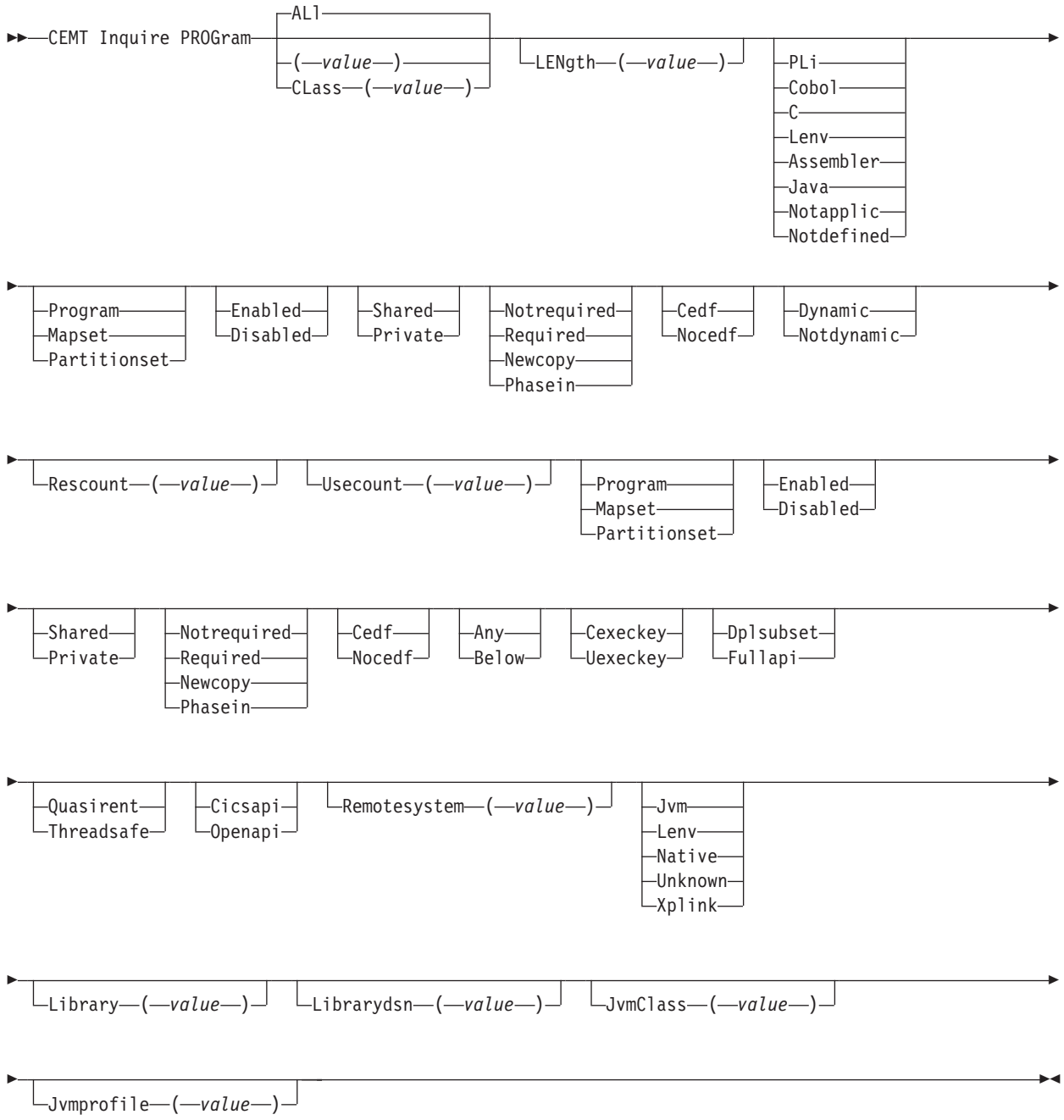
```
IN PROG
RESULT - OVERTYPE TO MODIFY
  Program(DFH$AALL)
  Length(0000002936)
  Language(Assembler)
  Progtype(Program)
  Status( Enabled )
  Sharestatus( Private )
  Copystatus( Notrequired )
  Cedfstatus( Cedf )
  Dynamstatus(Notdynamic)
  Rescount(000)
  Usecount(0000000)
  Dataloc(Below)
  Execkey(Uexeckey)
  Executionset( Fullapi )
  Concurrency(Threadsafe)
  Apist(Cicsapi)
  Remotesystem()
  Runtime( Jvm )
```

Figure 128. The expanded display of an individual entry (first panel)

```
IN PROG
RESULT - OVERTYPE TO MODIFY
  Program(DFH$AALL)
  Runtime( Jvm )
  Library(        )
  Librarydsn(                                          )
  Jvmclass( JavaApplClassFile                                        )
  Jvmclass(                                                          )
  Jvmclass(                                                          )
  Jvmclass(                                                          )
  Jvmclass(         )
  Jvmprofile(DFHJVMPR)
```

Figure 129. The expanded display of an individual entry (second panel)

CEMT INQUIRE PROGRAM



Displayed fields

Apist
displays the API attribute of the installed program definition. The 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 might run on whichever of the TCBs in use by CICS at the time is determined to be suitable.

Openapi
The program is not restricted to the CICS permitted application program interfaces. CICS executes 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).

Cedfstatus
displays what action the execution diagnostic facility (EDF) is to take for this program when the program runs under EDF. The values are:

Cedf When EDF is active, CEDF initiation and termination screens are shown by CEDF while this program is running. Other screens are also shown unless the program was translated using the NOEDF translator option.

Nocedf
All CEDF activities, including initiation and termination screens, stop while this program is being processed.

Concurrency
Displays the concurrency attribute of the specified program. The values are:

Threadsafe
The program is defined as being threadsafe, using appropriate serialization techniques to access shared resources. It does not need the automatic serialization provided by the CICS QR TCB, and can safely run on an open TCB.

Quasirent
The program is defined as not threadsafe, and requires the automatic serialization provided by QR TCB.

Copystatus
displays the load status of the specified program.

The values are:

Notrequired
The program status is such that you do not have to specify CEMT SET PROGRAM NEWCOPY or CEMT SET PROGRAM PHASEIN to make the program available. (If the program is already available for use and if you have put a new version in the DFHRPL or a dynamic LIBRARY concatenation, you need to issue a NEWCOPY or PHASEIN to replace the current version.)

Required
A search for the program failed during a load operation, and the program has been marked as not loadable to avoid the overhead of further load attempts.

You need to specify CEMT SET PROGRAM NEWCOPY or CEMT SET PROGRAM PHASEIN to enable the program to be loaded. However, before doing this, you should ensure that there is a copy of the program in the search order of LIBRARY concatenations (including DFHRPL and dynamic LIBRARYs).

You can reset this value by overtyping it. The values are:

Newcopy
CICS uses a new copy of the program if all the transactions currently using the program have finished (RESCOUNT equal to zero). If

RESCOUNT is not zero, NEWCOPY will fail. CICS loads the new version either from the search order of LIBRARY concatenations (including DFHRPL and dynamic LIBRARYs) or uses an LPA-resident version, depending on the PRIVATE or SHARED options. PRIVATE is the default setting.

Note: NEWCOPY cannot be specified for a program specified with the HOLD option, or for a Java program that runs in a JVM.

Phasein

CICS uses a new copy of the program now for all new transaction requests. CICS continues to use the old copy for all currently running transactions until they have finished (RESCOUNT equal to zero). CICS loads the new version either from the search order of LIBRARY concatenations (including DFHRPL and dynamic LIBRARYs) or uses an LPA-resident version, depending on the PRIVATE or SHARED options. PRIVATE is the default setting.

Note: PHASEIN cannot be specified for a program specified with the HOLD option, or for a Java program that runs in a JVM.

PHASEIN performs a REFRESH PROGRAM function to inform the loader domain that a new version of the program has been cataloged and that this version of the named program should be used in all future ACQUIRE requests.

Until the program goes to a zero RESCOUNT and a subsequent ACQUIRE PROGRAM has been performed, CEMT INQUIRE PROGRAM ( ) will return information on the first copy of the program.

Dataloc

displays whether the program is able to accept data addresses higher than 16MB. The values are:

Any The program can handle 31-bit addresses. The address of the data can be above or below the 16MB boundary. CICS returns an address above the 16MB boundary if possible.

Below

The program can handle only 24-bit addresses and must therefore be given addresses of data located below the 16MB boundary. If necessary, CICS copies data below the 16MB boundary before passing its address to the application program.

Dynamstatus

displays whether, if the program is the subject of a program link request, the request can be dynamically routed. The 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 EXEC CICS 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 EXEC CICS LINK command; otherwise it defaults to the local region.

For information about the dynamic routing of DPL requests, see in the *CICS Intercommunication Guide*.

Execkey
displays which access key the program is executing in. The values are:

Cexeckey
The program is executing in CICS access key, and has read and write access to both CICS-key and user-key storage.

Uexeckey
The program is executing in user access key, and has write access to user-key storage, but read-only access to CICS-key storage.

Executionset
displays whether the program is restricted to the API subset as for a distributed program link request. The values are:

Dplsubset
The program is restricted to the DPL API subset, as for a distributed program link request, when it runs in the local CICS region.

A program is always restricted to the DPL subset when it is invoked in a remote region via a DPL request, even if this option is not specified.

Fullapi
The program is not restricted to the DPL subset of the CICS API when it runs in the local CICS region, and can use the full API.

A program is always restricted to the DPL subset when it is invoked in a remote region via a DPL request, regardless of this option.

Note: You can reset this value by overtyping it with a different value.

Jvmclass
displays the name of the main class in the Java program to be given control by the JVM, as specified in the program definition, up to a maximum of 255 characters.

Jvmprofile
displays the name of the JVM profile that is to be used for the JVM in which this Java program runs.

Language
displays the language in which the program is written. The values are:

Pli, Cobol, C, Lenv, or Assembler
Supported languages. **Lenv** means that the module, whatever its language, exploits multi-language support, or was compiled with a Language Environment-conforming compiler.

Java The program is written in Java.

Notapplic
The language was not specified in the resource definition, and has not been loaded.

Notdefined
The language in which the program has been written is not defined.

Length(*value***)**
displays the size of the program in bytes. The value returned is 0000000000 if the program has not been loaded during this CICS run. No value is returned if the module is a remote program or a Java program that runs in a JVM.

Library(*value*)
displays the 8-character name of the LIBRARY from which the program is currently loaded.

Note:

- If the program was loaded from an installed LIBRARY, the Library and Librarydsn names will be returned.
- If the program was loaded from a LIBRARY that has been disabled, the Library name will be returned but the Librarydsn will be blank.
- If the program was loaded from a LIBRARY that has been discarded, both Library and Librarydsn will be blank.

Librarydsn(*value*)
displays the 44-character name of the data set within the LIBRARY from which the program is currently loaded.

Note:

- If the program was loaded from an installed LIBRARY, the Library and Librarydsn names will be returned.
- If the program was loaded from a LIBRARY that has been disabled, the Library name will be returned but the Librarydsn will be blank.
- If the program was loaded from a LIBRARY that has been discarded, both Library and Librarydsn will be blank.

Program(*value*)
indicates that this panel relates to a PROGRAM inquiry and displays an 8-character program identifier (1–8 characters), which is the name of a specific program entry in the table of installed program definitions.

Progtype
displays the type of program. The values are:

Program
The entry is defined in the CICS system definition as a program.

Mapset
The entry is defined in the CICS system definition as a map set.

Partitionset
The entry is defined in the CICS system definition as a partition set.

Remotesystem(*value*)
displays the 4-character name of the remote system in which the program is to execute.

Rescount(*value*)
displays a 3-character string identifying the number of separate invocations of this program that are taking place at the time of this inquiry. No value is returned if the module is a remote program or a Java program that runs in a JVM.

Note: This is not necessarily the same as the number of times the module has been loaded from DFHRPL or a dynamic LIBRARY concatenation.

Runtime
displays information about the runtime environment of the program. The values are:

Jvm The program is a Java program that runs in a Java Virtual Machine (JVM).

Lenv The program is running with Language Environment runtime support.

Native The program is running with its native language runtime support (not Language Environment).

Notknown
The runtime environment in not known because the program has not been loaded by CICS, therefore the source language cannot be 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
displays, when the next NEWCOPY or PHASEIN request is received, where the next new copy of the program is to be loaded from.

The values are:

Shared
CICS is using, or will use, a shared copy of the program from the link pack area (LPA). If no LPA version is available, the program is loaded from DFHRPL or a dynamic LIBRARY concatenation.

Private
The next new copy of the program is to be loaded from DFHRPL or a dynamic LIBRARY concatenation.

Note: You can reset this value by overtyping it with a different value.

Status
displays whether the program is available for use. The values are:

Enabled
The program is available for use.

Disabled
The program is not available for use.

Note: Programs beginning with “DFH” cannot be disabled because these characters are reserved for use by CICS.

Note: You can reset this value by overtyping it with a different value.

Usecount(*value***)**
displays a 10-character string identifying the total number of times the program has been executed since the start of the current CICS session. The use count is provided for all modules including Java programs, with the exception of remote programs. No value is returned if the program is remote.

CEMT INQUIRE REQUESTMODEL

Retrieves information about a REQUESTMODEL definition.

Description

INQUIRE REQUESTMODEL returns information about any currently installed request models.

Input

Press the Clear key to clear the screen, and type `CEMT INQUIRE REQUESTMODEL` (the minimum abbreviation is `CEMT I RE`). You get a display that lists the names and status of any request models.

(*value***)**
displays the 4–character field containing the (possibly generic) name of the destination CORBAServer for this REQUESTMODEL.

ALl
is the default.

Sample screen

```
 I RE
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Req(CFAC    ) Corbas(IIOP) Tra(CICF) Cor    Con Ign
  Req(III     ) Corbas(IIII) Tra(CIRP) Cor    Con Res
     Mod(*                         ) Int(*                        )


                                          SYSID=HA61 APPLID=CICSHA61
  RESPONSE: NORMAL                       TIME:  15.54.22  DATE: 09.03.98
PF 1 HELP       3 END       5 VAR       7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

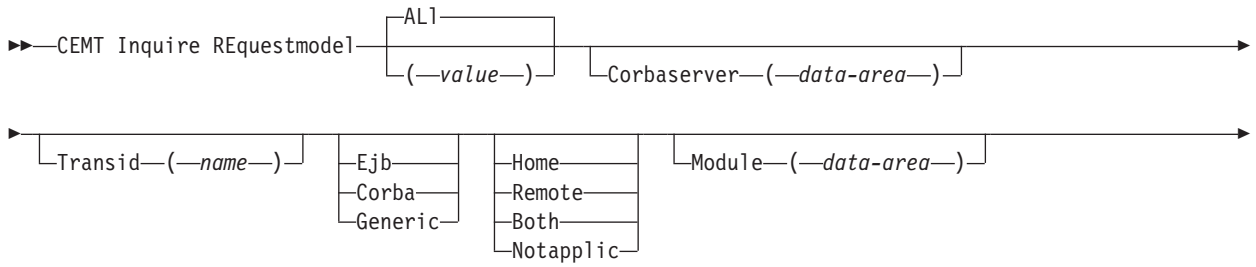
Figure 130. CEMT INQUIRE REQUESTMODEL screen

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format.

```
I RE
RESULT - OVERTYPE TO MODIFY
  All
  Requestmodel(DFHGFACT)
  Corbaserver
  Transid
  Type
  Intfacetype
  Module
  Interface
  Operation
  Beanname
```

Figure 131. The expanded display of an individual entry

CEMT INQUIRE REQUESTMODEL



Displayed fields

(value)
displays the 4–character field containing the (possibly generic) name of the destination CORBAServer for this REQUESTMODEL.

Beanname*(data-area)*
displays 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)*
displays the 4–character field containing the (possibly generic) name of the destination CORBAServer for this REQUESTMODEL.

Interface*(data-area)*
displays the 255-character field containing the (possibly generic) pattern matching the interface name for this REQUESTMODEL. This field is blank if the TYPE attribute is EJB.

Intfacetype
displays a value indicating the Java interface type for this REQUESTMODEL. The values are:

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)*
displays the 255-character field containing the (possibly generic) pattern defining the name scope of the interface and operation values for this REQUESTMODEL. This field is blank if the TYPE attribute is EJB.

Operation*(data-area)*
displays the 31-character field containing the pattern matching the IDL operation name for this REQUESTMODEL.

REquestmodel (*value***)**
displays the 8-character request model name.

Transid *(name)*
displays the 4-character name of the CICS transaction to be executed when a request matching the specification of the REQUESTMODEL is received.

Type
displays a value indicating the type of the REQUESTMODEL. The 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.

CEMT INQUIRE RRMS

Retrieve RRMS information.

Description

INQUIRE RRMS tells you if CICS does or does not accept inbound transactional EXCI work.

Input

Press the Clear key and type `CEMT INQUIRE RRMS` (the minimum abbreviation is `CEMT I R`).

You will get a display screen, a sample of which is shown below.

Sample screen

```
IN RRMS
STATUS:  RESULTS
 Rrm Ope
```

Figure 132. CEMT INQUIRE RRMS screen

CEMT INQUIRE RRMS

Displayed fields

Openstatus
displays a value indicating the the status of inbound transactional EXCI work. The values are:

Closed
indicates that CICS will not accept inbound transactional EXCI work.

Open indicates that CICS will accept inbound transactional EXCI work.

CEMT INQUIRE STATISTICS

Retrieve statistics information.

Description

INQUIRE STATISTICS tells you the interval during which statistics are accumulated, the end-of-day time for shutdown (or logical end-of-day) statistics, the time at which statistics are next recorded (that is, written to an SMF data set), and whether statistics recording is currently on or off.

For details of how to print statistics using the CICS-supplied utility program DFHSTUP, see the *CICS Operations and Utilities Guide*.

Input

Press the Clear key and type `CEMT INQUIRE STATISTICS` (the minimum abbreviation is `CEMT I ST`).

You will get a display screen, a sample of which is shown below. If you want to change any of the values shown, you can then tab to the highlighted or blank fields and overtype them as required.

Sample screen

```
IN STATI
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Sta On          Int( 030000 ) End( 000000 ) Nex(150000)
```

Figure 133. CEMT INQUIRE STATISTICS screen

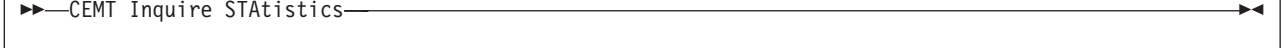
Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 134.

```
IN STAT
RESULT - OVERTYPE TO MODIFY
  Statistics
  Recording( Off )
  Record(              )
  Reset(              )
  Interval( 030000 )
  Endofday( 000000 )
  Nexttime(000000)
```

Figure 134. The expanded display

CEMT INQUIRE STATISTICS



Displayed fields

Endofday (*hhmmss***)**
displays the end-of-day time for recording statistics. At end-of-day, the current values of the CICS statistics counters are written out to an SMF data set and the statistics counters are reset.

End-of-day times must be in the range 000000-235959. The default end-of-day time is 12 midnight (000000).

Interval (*hhmmss*)
displays the length of time during which the statistics counters are incremented. At the end of each interval, the accumulated statistics are recorded and the statistics counters are reset.

Note: You can reset this value by overtyping it with a different value.

Interval times must be in the range 000100-240000. The default interval value is 030000. The interval value can be adjusted, but will have no effect unless the statistics recording status is set ON. You can set the statistics recording status ON or OFF using the RECORDING option on this command or by using the STATRCD system initialization parameter.

The arrival of the ENDOFDAY time always causes the current interval to be ended (possibly prematurely) and a new interval to be started. Only end-of-day statistics are recorded at the end-of-day time, even if it coincides exactly with the expiry of an interval.

Changing the ENDOFDAY value has an immediate effect on the times at which INTERVAL statistics are recorded. New interval expiry times are recalibrated in accordance with the new end-of-day time.

When you change the INTERVAL value or the ENDOFDAY value (and also when CICS is initialized), the length of the current (or first) interval is adjusted so that it expires after an integral number of intervals from the end-of-day time.

These rules are illustrated by the following example, where **I** indicates an interval recording, INTERVAL is set at 3-hourly intervals, CICS is initialized at 0815 hours, and **E** indicates the ENDOFDAY setting at 1700 hours.

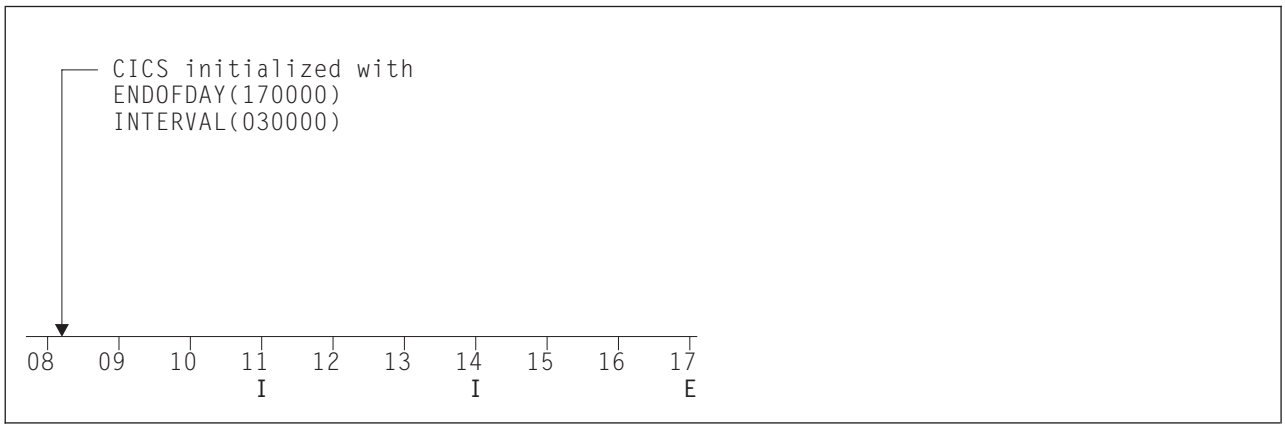


Figure 135. Changing the INTERVAL or ENDOFDAY value

Note: In this example, there would have been an interval at 0800 hours if CICS had been initialized earlier.

If you want your end-of-day recordings to cover 24 hours, set INTERVAL to 240000.

Nexttime (*hhmmss*)
displays the time at which statistics are next recorded and reset. This is the expiry time of the current interval or the end-of-day time, whichever is earlier.

Record (*input only field***)**
specifies whether existing statistics are to be written to an SMF data set. The value is:
Record
Existing statistics are to be written to an SMF data set. The statistics are those collected since the last ENDOFDAY or INTERVAL collection, or since the last time the RESETNOW function was used in this or the PERFORM STATISTICS command.

Note:

1. When both Rec(ord) and Res(et) are specified, statistics counters are written to the SMF data set and reset. The statistics that you are given are those collected since the last ENDOFDAY or INTERVAL collection, or since the last time the RESETNOW function was used in this or the PERFORM STATISTICS command.
2. RECORD RESET take effect only when the recording status is changed.

Recording
displays whether the recording of interval statistics is switched on or off. End-of-day statistics, requested statistics, and unsolicited statistics are always recorded, irrespective of the setting of the ON or OFF option. (Requested statistics are those called for by a CEMT PERFORM STATISTICS command or by an EXEC CICS PERFORM STATISTICS RECORD command.) The values are:
On Interval statistics are being recorded. Counters are incremented as usual, and are reset at the expiry of each interval and at end-of-day.
Off Interval statistics are not being recorded. Counters are still incremented as usual, and the recording of end-of-day, requested, and unsolicited statistics is still made.

Note: You can reset this value by overtyping it with a different value.

Reset (*input only field***)**
specifies whether the statistics counters are to be reset. The value is:
Reset
The statistics counters are to be reset. The reset takes effect only if the recording option is changed from OFF to ON or from ON to OFF. For information about the effect of reset on each individual counter, see the description of the CICS statistics tables in the *CICS Performance Guide*.

The action, for each counter, is one of the following:
- Not reset
- Reset to zero
- Reset to 1
- Reset to current value (this applies to peak values)
- An exception to the above.

STAtistics
indicates that this panel relates to a STATISTICS inquiry.

CEMT INQUIRE STREAMNAME

Retrieve information about currently connected MVS log streams.

Description

The INQUIRE STREAMNAME command allows you to inquire on the currently connected MVS log streams.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE STREAMNAME` (the minimum abbreviation is `CEMT I STR`). You get a display that lists the current status.
- Type `CEMT INQUIRE STREAMNAME` (`CEMT I STR`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i str fai sys`, the resulting display will show you the details of only those streamnames that are failed and that are the system log.

You can then tab to the highlighted or blank fields and overtype them with the required values.

(*value***)**
specifies an MVS system logger log stream name. The name can be a specific LSN of up to 26 characters in length.

ALl
is the default.

Sample screen

```
IN STRE
STATUS:  RESULTS
 Str(BELL.CICSHT61.DFHLOG      ) Use(0000001) Ok  Sys
```

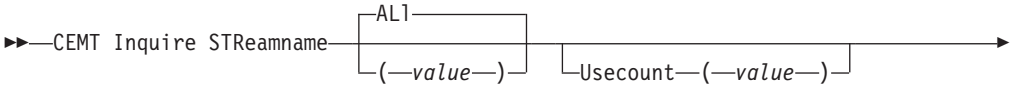
Figure 136. CEMT INQUIRE STREAMNAME screen

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 137.

```
IN STRE
RESULT
  Streamname(BELL.CICSHT61.DFHLOG)
  Usecount(0000001)
  Status(Ok)
  Systemlog(Syslog)
```

Figure 137. The expanded display of an individual entry

CEMT INQUIRE STREAMNAME



Displayed fields

Status
displays the status of the log stream. The values are:
Ok No errors have been detected
Failed
The MVS system logger has detected a problem with the specified log stream.

Streamname (*value*)
indicates that this panel relates to a STREAMNAME inquiry and displays an MVS system logger log stream name.

Systemlog
displays whether this log stream is the system log. The values are:
Syslog
The log stream is the system log.
Nosyslog
The log stream is not the system log.

Usecount (*value*)
displays the number of CICS journal names and forward recovery logs that are currently using the log stream.

The use count is always at least 1, because CICS does not maintain any information about a log stream that no longer has any users, in which case an INQUIRE STREAMNAME command returns an ERROR 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.

CEMT INQUIRE SYDUMPCODE

Retrieve information about the system dump code table.

Description

INQUIRE SYDUMPCODE allows you to see the current settings of the entries in the system dump table. 'NO' settings (NOSYSDUMP, for example) are shown as blank.

The INQUIRE command tells you:

- The system dump code or message identifier
- Whether or not CICS takes a system dump
- Whether the system dump request is local or related
- Whether or not CICS is to shut down following the dump
- The current number of dumps taken
- The maximum number of dumps
- Whether or not a dump produced by CICS is eligible for suppression by the MVS Dump Analysis and Elimination (DAE) component.

A system dump code is the CICS message number with the DFH prefix removed. Typically this leaves a 6-character code comprising 2 alphabetic characters and 4 numeric characters, for example AP0001. The command syntax permits up to 8 characters for the dump code and the dump codes you specify are padded with trailing spaces.

Each message has its own system default actions that you cannot necessarily override with the SET command. For example, when a serious error occurs, CICS shuts down even if you have asked for no shut down. See *CICS Messages and Codes* for information about the system action for each message.

There are other system parameters that take priority over your dump table entries. For example, EXEC CICS SET SYSTEM DUMPING NOSYSDUMP, or a DUMP=NO system initialization parameter inhibits system dumping. For information on using dumps, see the *CICS Problem Determination Guide*.

The INQUIRE SYDUMPCODE command returns entries that were added to the dump table by a user application program or CEMT command, and also entries that CICS added to the dump table through dump requests. Be aware that the entries that CICS added to the dump table are temporary, and they are not restored during a warm start, or during a cold or initial start. Entries added by a user application program or CEMT command are restored during a warm start, but not during a cold or initial start.

The transaction dump command SET TRDUMPCODE can also cause CICS to produce system dumps for the specified transaction dump code.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE SYDUMPCODE` (the minimum abbreviation is `CEMT I SYD`). You get a display that lists the current status.
- Type `CEMT INQUIRE SYDUMPCODE` (`CEMT I SYD`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i syd sy sh`, the resulting display will show you the details of only those system dump codes for which a system dump is to be taken and CICS is to be shut down.

You can then tab to the highlighted or blank fields and overtype them with the required values.

(*value***)**
is an 8-character system dump code. See *CICS Messages and Codes* for details of the codes.

ALl
is the default.

Sample screen

```
I SYD
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Syd(AP0001  ) Sys Loc     Max( 010 ) Cur(0000)     Dae
 Syd(AP0002  ) Sys Loc     Max( 999 ) Cur(0000)
```

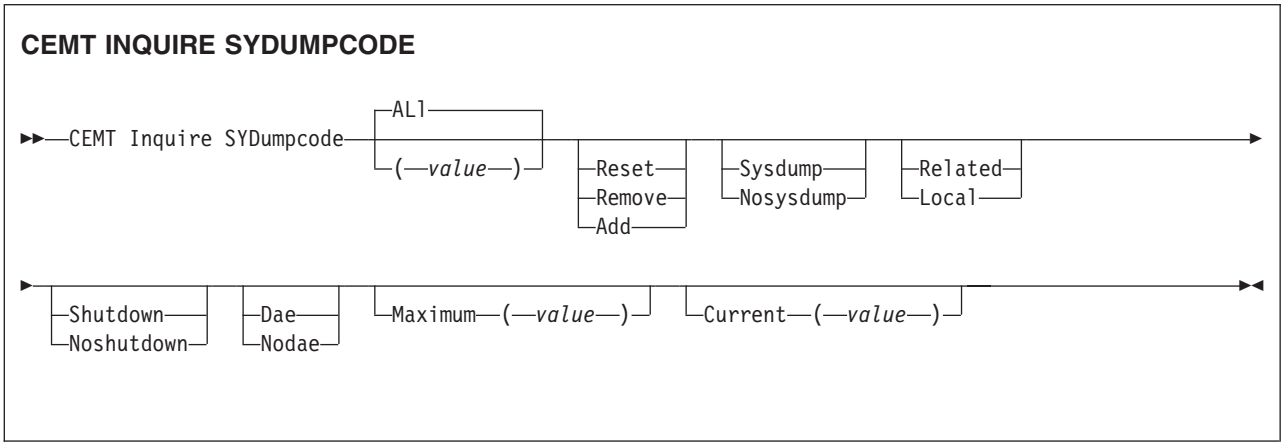
Figure 138. CEMT INQUIRE SYDUMPCODE screen

Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 139.

```
I SYD
RESULT - OVERTYPE TO MODIFY
  Sydumpcode(AP0001)
  Sysdumping( Sysdump )
  Dumpscope( Local )
  Shutoption( Noshutdown )
  Maximum( 010 )
  Current(0000)
  Action(              )
  Daeoption( Dae )
```

Figure 139. The expanded display



Displayed fields

Action (*input only field***)**
specifies the action to be taken with this system dump code. The values are:

Reset
The current number of calls for this dump code is reset to zero.

Remove
This system dump code is removed from the system dump code table.

Add This system dump code is added to the system dump code table.

Current (*value*)
displays the number of dump calls that have been made for this dump code since it was last reset.

Daeoption
displays whether a dump taken for CICS is eligible for suppression by the MVS dump analysis and elimination (DAE) component. The values are:

Dae A system dump is eligible for suppression by the DAE component.

Nodae
A system dump will not be suppressed by the DAE component. (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 *OS/390 MVS Diagnosis: Tools and Service Aids* manual.

Note: You can reset this value by overtyping it with a different value.

Dumpscope (*value*)
displays whether a system dump request is to be sent to MVS images in the sysplex which are running XCF/MRO connected CICS regions related to the CICS region on which the dump is initiated.

Note: A related CICS region is one in which the unit of work identifiers, in the form of APPC tokens, of one or more tasks match those in the CICS region that issued the dump request.

Related
A system dump request is to be sent for this system dump code.

Local A system dump request is not to be sent to other MVS images in the sysplex for this system dump code.

Note: You can reset this value by overtyping it with a different value.

Maximum (*value*)
displays the maximum number of dump calls for this system dump code which result in system dumps being taken. The value is in the range 0-999 and can be specified by using SYDUMAX in the SIT parameter. A value of 999 means the default, 'no limit'.

Note: You can reset this value by overtyping it with a different value.

Shutoption
displays whether CICS is to be shut down after the occurrence of an error corresponding to this dump code.

Shutdown
CICS is to be shut down.

Noshutdown
CICS is not to be shut down.

Note: You can reset this value by overtyping it with a different value.

Sydumpcode (*value*)
indicates that this panel relates to a SYDUMPCODE inquiry and displays an 8-character system dump code. See *CICS Messages and Codes* for details of the codes.

Sysdumping
displays whether a system dump is taken for this system dump code. The values are:

Sysdump
A system dump is taken for this system dump code.

Nosysdump
A system dump is not taken for this system dump code.

Note: You can reset this value by overtyping it with a different value.

CEMT INQUIRE SYSTEM

Retrieve information about CICS.

Description

INQUIRE SYSTEM returns information about the CICS system.

Input

Press the Clear key to clear the screen, and type `CEMT INQUIRE SYSTEM` (the minimum abbreviation is `CEMT I SYS`). You get a display that lists the current status. You can then tab to the highlighted or blank fields and overtype them with the required values.

To change various attributes, you can:

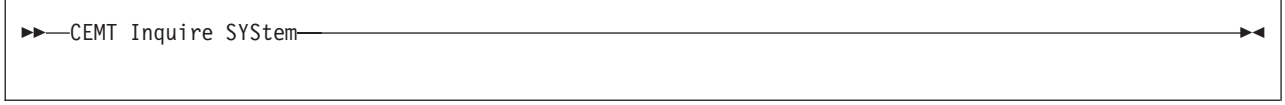
- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see Figure 140)
- Use the **CEMT SET SYSTEM** command.

Sample screen

```
 inq system
STATUS:  RESULTS - OVERTYPE TO MODIFY
  Aging( 32768 )                Progautoctlg( Ctlgmodify )
  Akp( 04000 )                  Progautoexit( DFHPGADX )
  Cicstslevel(030100)           Progautoinst( Autoactive )
  Cmdprotect(Nocmdprot)         Reentprotect(Reentprot)
  Db2conn()                     Release(0650 )
  Debugtool( Nodebug )          Runaway( 0005000 )
  Dfltuser(WALSH)               Scandelay( 0500 )
  Dsalimit( 07340032 )          Sdtran(CESD)
  Dsrtprogram( NONE     )       Sosabovebar(Notsos)
  Dtrprogram( DFHDYP    )       Sosaboveline(Notsos)
  Dumping( Sysdump )            Sosbelowline(Notsos)
  Edsalimit( 0268435456 )       Storeprotect(Active)
  Forceqr( Noforce )            Time( 0001000 )
  Logdefer( 00030 )             Tranisolate(Active)
  Maxtasks( 075 )
  Memlimit(Nolimit)
  Mrobatch( 001 )
  Oslevel(010400)


                                          SYSID=HT63 APPLID=CICSHT63
 RESPONSE: NORMAL                       TIME:  16.00.11  DATE: 10.03.01
PF 1 HELP       3 END       5 VAR       7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 140. CEMT INQUIRE SYSTEM screen

CEMT INQUIRE SYSTEM



Displayed fields

Aging (*value***)**
displays the time factor for CICS to use in the priority aging algorithm for incrementing the priority of a task. The value is expressed as “milliseconds per unit of priority”, and is in the range 0–65535.

The priority aging factor is defined initially by the **PRTYAGE** system initialization parameter, in which the value represents the number of milliseconds that must elapse before the priority of a waiting task can be adjusted upward by 1. See PRTYAGE in the *CICS System Definition Guide* for more information.

Note: You can reset this value by overtyping it with a different value.

Akp (*value***)**
displays the activity keypoint trigger value, which is the number of logging operations between the taking of keypoints. It is in the range 200–65535. A value of zero indicates that there is no activity keypointing.

Note: You can reset this value by overtyping it with a different value.

Cicstslevel (*value***)**
displays a 6-character value identifying the version, release and modification level of the CICS Transaction Server for z/OS product under which the CICS region is running. The value is of the form *vvrrmm*, and CICS Transaction Server Version 3 Release 2 returns 030200.

Cmdprotect (*value***)**
displays whether command protection, which validates start addresses passed on in CICS commands, is active or not (that is, whether the CMDPROT system initialization parameter specifies YES or NO). The values are:

Cmdprot
Command protection is active. CICS checks to ensure that the task itself has write access to the storage referenced on the command before writing to the storage on the task’s behalf.

Nocmdprot
Command protection is not active. CICS does not check to ensure that the task itself has write access to the storage referenced on the command before writing to the storage on the task’s behalf.

Db2conn (*value***)**
displays the name of the currently installed DB2CONN.

Debugtool(*value***)**
displays 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 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 debugging tool.

For more information about debugging profiles, see the *CICS Application Programming Guide*.

Dfltuser (*value***)**
displays the userid of the default user for this instance of the CICS region.

Dsalimit (*value***)**
displays the maximum amount of storage, as a total number of bytes, within which CICS can dynamically allocate storage for the four individual DSAs that reside below the 16MB boundary. (For a description of this parameter, see the *CICS System Definition Guide*.)

Note: You can reset this value by overtyping it with a different value.

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 dynamic storage is freed in the individual DSAs.

Dsrtprogram (*pgrmid***)**
displays the name of the distributed routing program.

Note: You can reset this value by overtyping it with a different value.

Dtrprogram (*pgrmid***)**
displays the name of the dynamic routing program.

Note: You can reset this value by overtyping it with a different value.

DUMPING(*value***)**
displays whether CICS system dumps are suppressed. The values are:
NOSYSDUMP
System dumps are suppressed.
SYSDUMP
System dumps are not suppressed.

These values are set by the SIT parameter DUMP=YES or NO.

Edsalimit (*value***)**
displays the maximum amount of storage, as a total number of bytes, within which CICS can dynamically allocate storage for the four individual DSAs that reside above the 16MB boundary. (For a description of this parameter, see the *CICS System Definition Guide*.)

Note: You can reset this value by overtyping it with a different value.

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 dynamic storage is freed in the individual DSAs.

Forceqr (*value*)
displays whether CICS is to CICSAPI force user application programs that are defined as threadsafe to run on the quasi-reentrant (QR) TCB. The 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.

Nforce
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.

Logdefer (*value*)
displays the log deferral interval in milliseconds, 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. It is in the range 0-65535. See the *CICS System Definition Guide* for information about the Logdefer parameter and associated system initialization parameter **LGDFINT**.

Note: You can reset this value by overtyping it with a different value.

Maxtasks (*value*)
displays the maximum number of tasks, both active and suspended, allowed at any one time in the CICS system. It must have a value in the range 1–999.

Note:

1. You can reset this value by overtyping it with a different value.
2. The value assigned to MAXTASKS might be less than the requested value, because of CICS storage constraints. If this occurs, the message CEILING REACHED is displayed when the request is made.

Memlimit (*value*)
displays the amount of storage available above the 2GB boundary (above the bar), for use by the CICS region. A value of NOLIMIT indicates that no limit has been imposed on the amount of storage that the region can attempt to use

Mrobatch (*value*)
displays the number of MRO requests from connected regions which are to be batched before this region is posted. It is a value in the range 1–255.

Note: You can reset this value by overtyping it with a different value.

Oslevel (*value*)
displays 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 is shown as 020400.

z/OS Version 1 Release 4 Modification 0 appears as 010400

Progautoctlg
displays whether autoinstalled program definitions are to be cataloged. The values are:

Ctlgall
All autoinstalled program definitions are to be cataloged and restored on a warm or emergency start.

CtIgmodify
Autoinstalled program definitions are to be cataloged only if they are modified (for example, by a CEMT SET PROGRAM command), so that the modified definitions are restored on a warm or emergency restart.

CtIgnone
No autoinstalled program definitions are to be cataloged. They are autoinstalled again after a warm or emergency start.

Note: You can reset this value by overtyping it with a different value.

Progautoexit (*value*)
displays the name of the user-provided program that is called by the program autoinstall code to select or modify a model definition.

Note: You can reset this value by overtyping it with a different value.

Progautoinst
displays whether autoinstall for programs is active or inactive. The values are:

Autoactive
Autoinstall for programs is active. On first use, if a program, mapset, or partitionset is not defined, the definition is created dynamically.

Autoinactive
Autoinstall is not active. If a program is not defined, a PGMIDERR or transaction abend occurs when it is referenced.

Note: You can reset this value by overtyping it with a different value.

Reentprotect (*value*)
displays whether read-only storage is in use for reentrant programs (that is, whether the RENTPGM system initialization parameter specifies PROTECT or NOPROTECT). The values are:

Noreentprotect
CICS allocates storage for the read-only DSAs (RDSA and ERDSA) from CICS-key storage. Reentrant programs do not have the protection of residing in read-only storage, and can be modified by programs executing in CICS key.

Reentprotect
CICS allocates storage for the RDSA and ERDSA from key-0, non-fetch protected, storage. CICS loads reentrant programs into this storage, and the programs are protected by residing in read-only storage.

Release (*value*) (supported for compatibility only)
displays a 4-digit number (value 0650 for this release of CICS TS) representing the level of the CICS code running this region.

This field is maintained for compatibility with previous releases of CICS only. As an exclusive element of CICS Transaction Server for z/OS, CICS does not have a product version and release number of its own. To determine the release number of the product, see the value displayed for CICSTSLEVEL.

Runaway (*value*)
displays the interval, in milliseconds, for which a task can have control before it is assumed to be looping.

Note: You can reset this value by overtyping it with a different value.

If you are resetting this value, you can specify 0, or a value in the range 500–2700000. The value you specify is rounded down to a multiple of 500.

CICS purges a task if the task has not given up control after this interval (that is, CICS assumes that the task is looping). If you specify zero, runaway task control is inoperative (that is, tasks do not get purged if they appear to be looping). The value is independent of, and can be less than, the value in the TIME operand.

CICS runaway-task detection is based upon task time (that is, the interval is reset each time a task receives control of the processor, and is then decremented while the task is in control). You do not, therefore, need to allow for the effect of other jobs when setting the runaway value.

Scandelay (*value***)**
displays the maximum number of milliseconds between the receipt of a VTAM request, and the time when CICS begins to process the request. The number is in the range 0–5000, and is less than or equal to the TIME value.

Note:

1. You can reset this value by overtyping it with a different value.
2. In the summary report of the dispatcher statistics, SCANDELAY is referred to as ICVTSD.

Sdtran (*value***)**
displays 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.

Sosabovebar(*value***)**
displays whether CICS is short on storage in the dynamic storage areas above the 2GB boundary (above the bar).

Notsos
CICS is not short on storage in any of the dynamic storage areas above the 2GB boundary.

Sos CICS is short on storage in at least one of the dynamic storage areas above the 2GB boundary.

Sosaboveline(*value***)**
displays whether CICS is short on storage in the dynamic storage areas above the 16MB line, but below the 2GB boundary.

Notsos
CICS is not short on storage in any of the dynamic storage areas above the 16MB line (but below the 2GB boundary).

Sos CICS is short on storage in at least one of the dynamic storage areas above the 16MB line (but below the 2GB boundary).

Sosbelowline(*value***)**
displays whether CICS is short on storage in the dynamic storage areas below the 16MB line.

Notsos
CICS is not short on storage in any of the dynamic storage areas below the 16MB line.

Sos CICS is short on storage in at least one of the dynamic storage areas below the 16MB line.

Storeprotect (*value***)**
displays whether storage protection is active in the CICS region. The values returned are:

Active
CICS is operating with the CICS storage protection facility.

Inactive
CICS is operating without the CICS storage protection facility.

See the description of the *CICS System Definition Guide* for background information about CICS storage protection.

Time (*value***)**
displays the interval, in milliseconds, for which CICS releases control to the operating system if no transactions are ready to resume processing. This interval is known as the 'region exit interval'.

Note:

1. You can reset this value by overtyping it with a different value. It can have a value in the range 100–3600000, and must be greater than or equal to the SCANDELAY value.
2. In the summary report of the dispatcher statistics, TIME is referred to as ICV.

Tranisolate
displays the status of transaction isolation. The values returned are:
Active
Transaction isolation is active in the CICS region.
Inactive
CICS is running without transaction isolation, either because the support is not available, or it was not requested at CICS initialization.

CEMT INQUIRE TASK

Retrieve information about a user task.

Description

INQUIRE TASK returns information about user tasks. Only information about user tasks can be displayed or changed; information about CICS-generated system tasks or subtasks cannot be displayed or changed. System tasks are those tasks started (and used internally) by CICS, and not as a result of a user transaction.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE TASK` (the minimum abbreviation is `CEMT I TA`). You get a display that lists the current status.
- Type `CEMT INQUIRE TASK` (`CEMT I TA`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i ta i`, the resulting display will show you the details of only those tasks for which the data is not shared with other tasks (isolated).

To change various attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see "Overtyping a display" on page 128).
- Use the CEMT SET TASK command.

(*value***)**
is the CICS-generated task number, in the range 1–99999.

ALl
is the default. The maximum number of tasks displayed is 32000.

TClass (*value***)**
is the 8-character transaction class name to which the transaction belongs. The maximum number of tasks displayed is 32000.

You cannot specify a list of identifiers, nor can you use the symbols * and + to specify a family of tasks.

Sample screen

```
IN TASK
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Tas(0000026) Tra(CEMT) Fac(S262) Sus Ter Pri( 255 )
    Sta(TO) Use(CICSUSER) Uow(B0C9D8D111440007) Hty(ZCIOWAIT)
 Tas(0000030) Tra(CEMT) Fac(S263) Run Ter Pri( 255 )
    Sta(TO) Use(CICSUSER) Uow(B0C9D8E9C3B1FBO9)
```

Figure 141. CEMT INQUIRE TASK screen

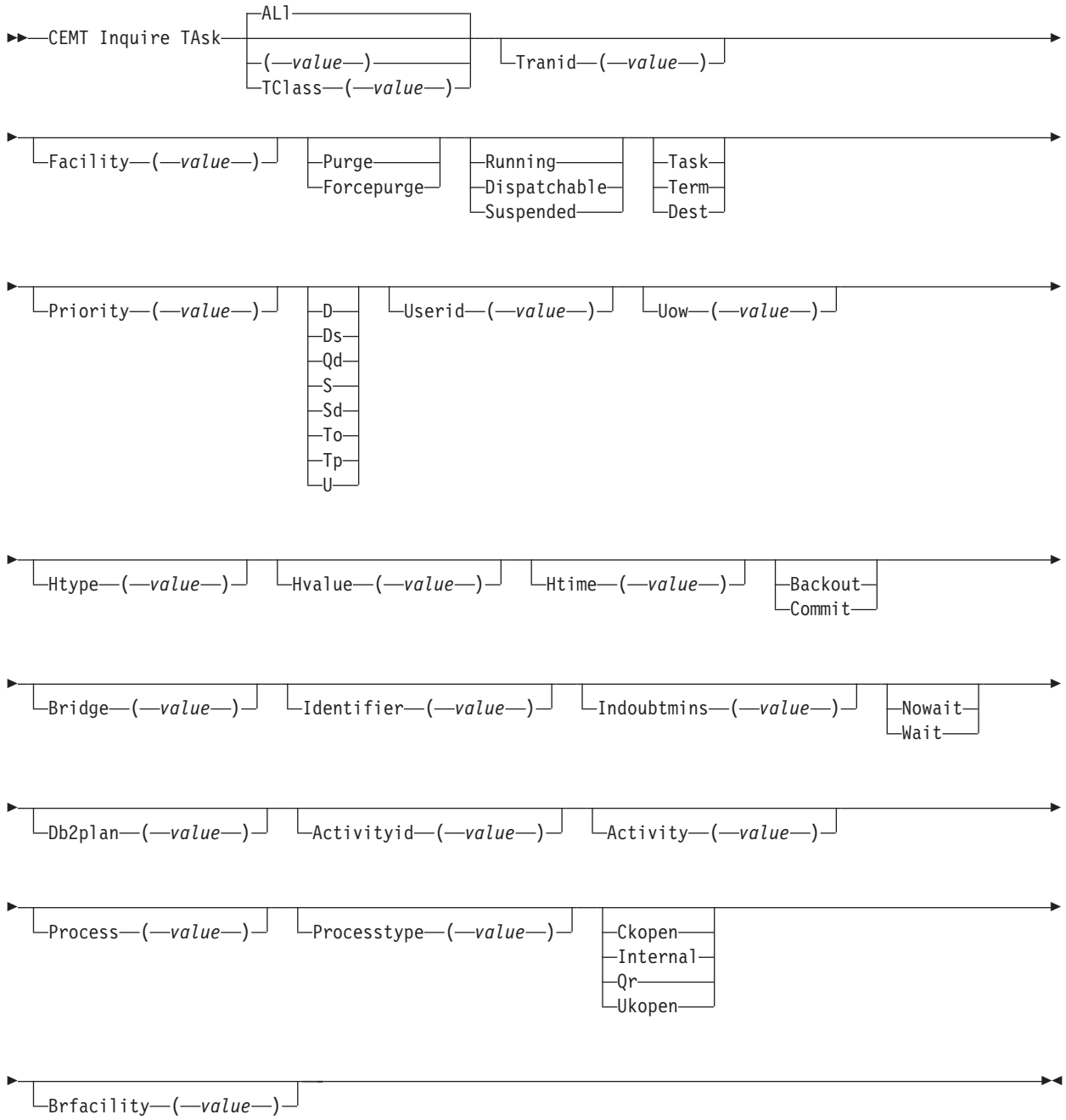
Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 142 on page 282.

```
 I TASK
 RESULT - OVERTYPE TO MODIFY
   Task(0000026)
   Tranid(CEMT)
   Facility(S262)
   Runstatus(Suspended)
   Ftype(Term)
   Priority( 255 )
   Purgetype(              )
   Startcode(TO)
   Userid(CICSUSER)
   Uow(B0C9D8D111440007)
   Htype(ZCIOWAIT)
   Hvalue(DFHZARQ1)
   Htime(000159)
   Indoubt(Backout)
   Indoubtwait(Wait)
   Bridge()
   Identifier()
   Indoubtmins(000000)
   Db2plan()
   Activityid()
   Activity()
   Process()
   Processtype()
   Tcb(QR)
   Brfacility(0000000000000000)

                                                SYSID=JOHN APPLID=I
                                              TIME:  11.26.40  DATE: 08
PF 1 HELP 2 HEX 3 END       5 VAR        7 SBH 8 SFH       10 SB 11 SF
```

Figure 142. The expanded display of an individual entry

CEMT INQUIRE TASK



Displayed fields

Activity *(value)*
displays the 16-character, user-assigned, name of the CICS business transaction services activity that this task is executing on behalf of.

Activityid *(value)*
displays the 52-character, CICS-assigned, identifier of the CICS business transaction services activity that this task is executing on behalf of.

Brfacility(*value*)
displays the 8-byte facility token representing the virtual terminal used by the current task if it is used by the 3270 bridge mechanism. If the task is not currently running in the 3270 bridge environment, zeroes are returned.

Bridge (*value*)
Displays the 4-character name of the bridge monitor transaction if the current task is running in a 3270 bridge environment, and was started by a bridge monitor transaction with a START BREXIT TRANSID command. Otherwise, blanks are returned.

Db2plan (*value*)
displays the 1- to 8- character name of the DB2 plan being used by this task, or blanks if no DB2 plan is being used.

Facility (*value*)
displays a 4-character string identifying the name of the terminal or queue that initiated the task. If no FACILITY value is displayed, the task was started without a facility.

Ftype
displays the type of facility that initiated this task. The values are:

Task The task was initiated from another task.
Term The task was initiated from a terminal.
Dest The task was initiated by a destination trigger level as defined in the TDQUEUE resource definition.

Htime (*value*)
displays the time (in seconds) that the task has been in the current suspended state.

Htype (*value*)
displays the reason why the task is suspended. A null value indicates that there is no hold-up, except for the necessity of reaching the head of the queue.

Hvalue (*value*)
displays a 16-character resource name, such as a file name, or a value such as a TCLASS value.

For information on the values that can appear in the HTYPE and HVALUE 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*.

Indoubtmins (*value*)
displays 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.

Indoubt (*value*)
displays the action (based on the ACTION attribute of the TRANSACTION resource definition) 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 Indoubtwait expires.

The 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.

Indoubtwait (*value***)**
displays how (based on the WAIT attribute of the TRANSACTION definition) a unit of work (UOW) is to respond if a failure occurs while it is in an in-doubt state. The 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.

For further information about the meaning of the ACTION and WAIT attributes of the TRANSACTION definition, see the *CICS Resource Definition Guide*.

Identifier (*value***)**
returns a 48-character field containing user data provided by the bridge exit, if the task was initiated in the 3270 bridge environment, or blanks, otherwise. This field is intended to assist in online problem resolution.

For example, it could contain the Websphere MQ™ correlator for the CICS-MQ bridge, or a Web token.

Priority (*value***)**
displays the priority of the task, in the range 0–255 where 255 is the highest priority.

Note: You can reset this value by overtyping it with a different value.

Process *(value)*
displays the 36-character name of the CICS business transaction services process that this task is executing on behalf of.

Processtype *(value)*
displays the 8-character process-type of the CICS business transaction services process that this task is executing on behalf of.

Purgetype (*input only field***)**
specifies whether a task is to be purged or forcepurged. The values are:
Purge
The task is to be terminated. Termination occurs only when system and data integrity can be maintained.
Forcepurge
The task is to be terminated immediately. System integrity is not guaranteed. In some extreme cases, for example if a task is forcepurged during backout processing, CICS terminates abnormally. If you want to terminate a task but do not want to terminate CICS, you should use PURGE instead of FORCEPURGE.

Runstatus
displays the status of this task. The values are:
Running
The task is running.
Dispatchable
The task is dispatchable.

Suspended
The task is suspended.

Startcode (*value*)
displays how this task was started. The values are:

D A distributed program link (DPL) request. The program cannot issue I/O requests against its principal facility or any syncpoint requests.

Ds A distributed program link (DPL) request, as for code D, with the exception that the program can issue syncpoint requests.

Qd A transient data trigger level was reached.

S Start command (no data)

Sd Start command (with data)

To The operator typed a transaction code at the terminal.

Tp The transaction was started by presetting the transaction ID for the terminal.

U User-attached task.

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).

Task (*value*)
indicates that this panel relates to a TASK inquiry and displays a CICS-generated task number in the range 1–99999.

Tcb (*value*)
displays the type of TCB under which the task is running. The values are:

Ckopen
The task is running under a CICS key open TCB (for example, a J8 mode TCB, used for JVMs in CICS key).

Internal
The task is running under one of the CICS internal TCBs. An internal TCB can be one of the following:

- The concurrent mode (CO) TCB
- The file-owning mode (FO) TCB
- The shared class cache (JM) TCB
- The resource-owning mode (RO) TCB
- The ONC/RPC mode (RP) TCB
- The sockets listener mode (SL) TCB
- The secure sockets layer mode (SO) TCB
- A sockets mode (S8) TCB
- The FEPI mode (SZ) TCB.

Qr The task is running under the CICS QR TCB.

Ukopen
The task is running under a user key open TCB (for example, a J9 mode TCB, used for JVMs in user key).

Tranid (*value*)
displays a 4-character string identifying the transaction name associated with the task.

Uow (*value*)
displays the 16-character local identifier of the unit of work associated with this task.

Userid (*value*)
displays the user currently associated with the task.

CEMT INQUIRE TCLASS

Retrieve information about tasks within a task class.

Description

INQUIRE TCLASS returns information about the current and maximum number of tasks, the purge threshold, and the status of a user-defined task class.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE TCLASS` (the minimum abbreviation is `CEMT I TCL`). You get a display that lists the current status.
- Type `CEMT INQUIRE TCLASS` (`CEMT I TCL`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i tc p(value)`, the resulting display will show you the details of only those transaction classes that have a purge threshold of the value indicated.

To change various attributes, use the CEMT SET TCLASS command.

(*value*)
is the 8-character transaction class name.

ALl
is the default.

Sample screen

```
 IN TCLASS
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Tcl(DFHTCIND) Max( 010 ) Act(000) Pur( 0000000 )
     Que(000000)
  Tcl(DFHTCL01) Max( 001 ) Act(000) Pur( 0000000 )
     Que(000000)
  Tcl(DFHTCL02) Max( 001 ) Act(000) Pur( 0000000 )
     Que(000000)
  Tcl(DFHTCL03) Max( 001 ) Act(000) Pur( 0000000 )
     Que(000000)
  Tcl(DFHTCL04) Max( 001 ) Act(000) Pur( 0000000 )
     Que(000000)
  Tcl(DFHTCL05) Max( 001 ) Act(000) Pur( 0000000 )
     Que(000000)
  Tcl(DFHTCL06) Max( 001 ) Act(000) Pur( 0000000 )
     Que(000000)
  Tcl(DFHTCL07) Max( 001 ) Act(000) Pur( 0000000 )
     Que(000000)
+ Tcl(DFHTCL08) Max( 001 ) Act(000) Pur( 0000000 )
     Que(000000)
```

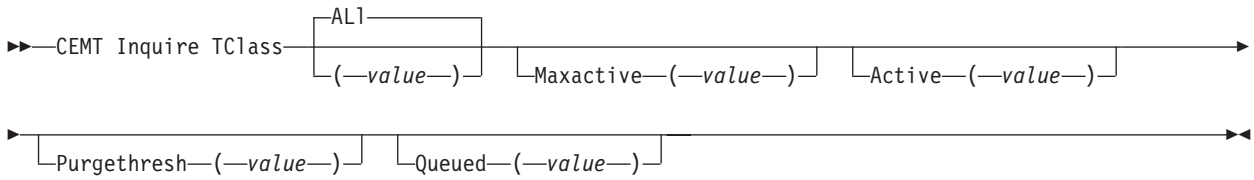
Figure 143. CEMT INQUIRE TASK screen

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 144 on page 288.

```
IN TCLASS
RESULT - OVERTYPE TO MODIFY
  Tclass(DFHTCIND)
  Maxactive( 010 )
```

Figure 144. The expanded display of an individual entry

CEMT INQUIRE TCLASS



Displayed fields

Active (*value***)**
displays the total number of transactions that are currently active in a user-defined transaction class.

Maxactive (*value***)**
displays the largest number of transactions in the transaction class which are allowed to run concurrently. The value can be in the range 0-999.

Note: You can reset this value by overtyping it with a different value.

Purgethresh (*value***)**
displays the limit at which the queuing transactions are purged for the transaction class. It can have a value in the range 0000000–1000000. Zero means that the transactions are not purged. A nonzero number “n” means that the first (n-1) transactions to arrive are queued and the nth is purged.

Note: You can reset this value by overtyping it with a different value.

Queued (*value***)**
displays the total number of current tasks that are suspended because the class maximum has been reached.

TClass (*value***)**
indicates that this panel relates to a TCLASS inquiry and displays an 8-character transaction class name.

CEMT INQUIRE TCPIP

Inquire about CICS internal TCP/IP support status.

Description

INQUIRE TCPIP returns information about the current status of CICS internal TCP/IP support.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE TCPIP` (the minimum abbreviation is `CEMT I TCPIP`). You get a display that lists the current status.
- Type `CEMT INQUIRE TCPIP` (`CEMT I TCPIP`) followed by as many of the other attribute settings that you wish to view.

To change various attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see "Overtyping a display" on page 128).
- Use the CEMT SET TCPIP command.

Displayed fields

Actsockets(*value*)
Displays the current number of active IP sockets managed by the CICS sockets domain.

Crlprofile(*value*)
Displays the name of the profile that authorizes CICS to use the certificate revocation lists in an LDAP server.

Maxsockets (*value*)
Displays the maximum number of IP sockets that can be managed by the CICS sockets domain.

Openstatus
Displays the status of CICS internal TCP/IP support. The values are:

Open CICS internal sockets 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(*value*)
Displays whether CICS is configured to use local or sysplex caching of session ids. The values are:

CICS CICS is configured to cache session ids in the local CICS region.

SYSPLEX
CICS is configured to cache session ids in the coupling facility.

CEMT INQUIRE TCPIPSERVICE

Retrieve information about TCP/IP ports on which CICS internal TCP/IP support is currently listening on behalf of other CICS services.

Description

INQUIRE TCPIPSERVICE returns information about the state of a service using CICS internal TCP/IP support.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE TCPIPSERVICE` (the minimum abbreviation is `CEMT I TCPIPS`). You get a display that lists the current status.
- Type `CEMT INQUIRE TCPIPSERVICE` (`CEMT I TCPIPS`) followed by as many of the other attribute settings that you wish to view.

To change various attributes, you can:

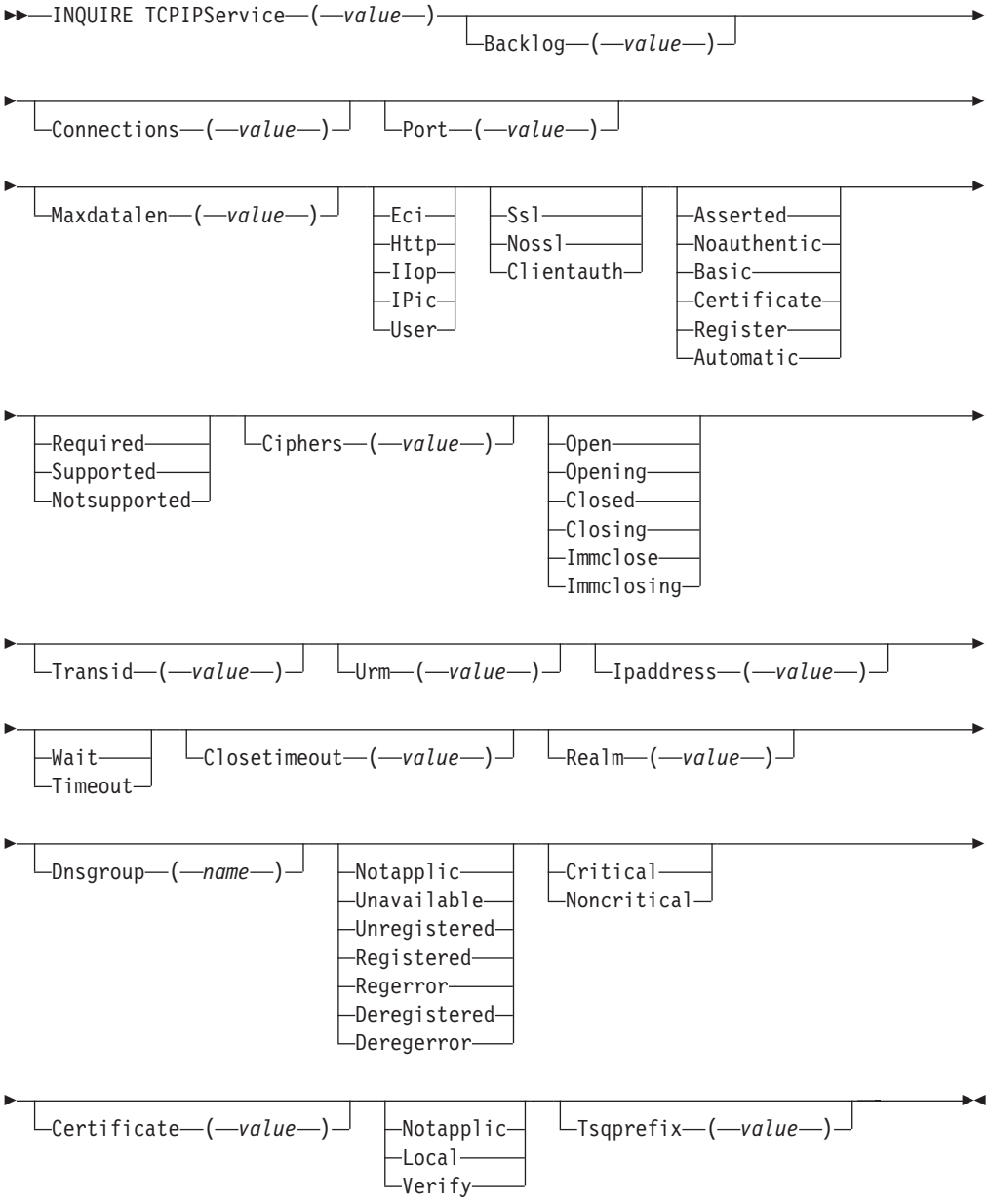
- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 128).
- Use the CEMT SET TCPIPSERVICE command.

Sample screen

```
inq tcpipservice
RESULT - OVERTYPE TO MODIFY
  Tcpipservice(HTTPNSSL)
  Backlog( 00005 )
  Connections(00000)
  Port(20130)
  Maxdatalen( 000032 )
  Protocol(Http)
  Ssltype(Ssl)
  Authenticate(Noauthentic)
  Privacy(Supported)
  Ciphers(0504352F0A0903060201)
  Openstatus( Open )
  Transid(CWXN)
  Urm( DFHWBADX )
  Ipaddress(9.20.101.6)
  Socketclose(Timeout)
  Closetimeout(000005)
  Realm(Salary System)
  Dnsgroup()
  Dnsstatus(              )
  Grpcritical(Noncritical)
  Certificate()
  Attachsec()
  Tsqprefix()
```

Figure 145. CEMT INQUIRE TCPIPSERVICE screen

CEMT INQUIRE TCPIPSERVICE



Displayed fields

Attachsec
indicates the level of attach-time security used by the connection. Values are:

Local
CICS does not require a userid or password from clients.

Notapplic
This option has no meaning for Web interface or IIOP TCP/IP services.

Verify
Incoming attach requests must specify a user identifier and a user password.

Authenticate
indicates the scheme used to authenticate clients. Values are:

Asserted
Asserted identity authentication is used.

This value is returned only when PROTOCOL has a value of IIOP.

Basic
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.

Certificate
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.

Register
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.

Automatic
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.

For more information about authentication and identification of HTTP and IIOP clients, see the *CICS RACF Security Guide*.

Backlog (*value***)**
Change the maximum number of requests which can be queued in TCP/IP waiting to be processed by the service.

Certificate (*value***)**
specifies the label of an X.509 certificate that is to be used in the SSL handshake for the TCP/IP service. Certificate labels can be up to 32 bytes, and are specified for the CICS region's userid within a key ring defined in the external security manager's database (for example, as defined by the RACDCERT command). If this attribute is omitted, the default certificate defined in the key ring for the CICS region userid is used.

Ciphers(*value***)**
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.

Closetimeout (*value*)
returns a fullword value containing the number of seconds specified for the timeout period. This can be 0 through 86400 (24 hours). For the HTTP protocol, 0 should not be specified, because this setting means that persistent connections cannot be maintained.

Connection (*value*)
The number of current sockets connections for this service

Dnsgroup(*name*)
returns the 18–character Domain Name System (DNS) group name that this TCPIPSERVICE registers with OS/390 Workload Manager (WLM).

Dnsstatus
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 z/OS.

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.

Grpcritical
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 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 (*value*)
returns an IP address. If you specify an IP address in the TCPIPSERVICE definition, that address is returned, otherwise the default IP address is returned. If there is more than one IP address on a z/OS system, only the default is returned.

Maxdatalen (*value*)
returns a fullword value containing the maximum length of data that may be received by CICS as an HTTP server. This can be 3 KB through 524288 KB. The default is 32 KB.

Openstatus
returns the state of the service. Values are:

Open CICS internal sockets support is open.

Opening
CICS internal sockets support is in the process of opening.

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.

Immclose
'CICS internal sockets support has immediately terminated for this service.

Immclosing
CICS internal sockets support is in the process of immediate termination.

Port (*value*)
returns the number of the port on which CICS is listening on behalf of this service.

Privacy
returns the level of SSL encryption required for inbound connections to this service. 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
identifies to CICS the type of service to be provided on the TCP/IP port. Values are:

Eci
Connections are handled by CICS ECI over TCP/IP support.

Http
Connections are handled by CICS Web support.

IIop
Connections are handled by CICS IIOP support. IIOP is required for TCPIPSERVICEs that are to accept inbound requests for enterprise beans.

IPic
IP interconnectivity.

User
The user-defined protocol is used. Requests are passed to the analyzer program for the TCPIPSERVICE, and handled using CICS Web support facilities, but the HTTP specifications are not used to check the messages.

Realm (*value***)**
returns the 56-character realm that is used during the process of HTTP basic authentication.

Socketclose
returns a CVDA value indicating whether a timeout value is in effect for the TCPIPSERVICE. 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.

Ssltype
returns a CVDA value indicating the status of SSL support for this service. CVDA values are

Ssl
Non client authentication SSL flows are processed on this port

Nossl
Non SSL flows are processed on this port

Clientauth
SSL client authentication flows are processed on this port

TCPIPService (*value***)**
8-character name identifying this service.

Transid (*value***)**
4-character transaction id used on the attach for the task started to process a new request.

Tsqprefix (*value***)**

is no longer required or used in CICS Transaction Server for z/OS, Version 3 Release 2 and later releases.

Urm (*value***)**
The 8-character name of the user-replaceable program to be invoked by the attached task.

CEMT INQUIRE TDQUEUE

Retrieve information about transient data queues.

Description

INQUIRE TDQUEUE returns information about a named transient data queue that is defined in a TDQUEUE resource definition.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE TDQUEUE` (the minimum abbreviation is `CEMT I TD`). You get a display that lists the current status.
- Type `CEMT INQUIRE TDQUEUE` (`CEMT I TD`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i td ind en`, the resulting display will show you the details of only those transient data queues that are indirect and enabled.

To change various attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see "Overtyping a display" on page 128).
- Use the CEMT SET TDQUEUE command.

(*value***)**
is the identifier (1-4 characters) of a transient data queue.

All
is the default.

Sample screen

```
 IN TDQ
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Tdq(CADL)              Ind Nam(CSSL)
  Tdq(CAIL)              Ind Nam(CSSL)
  Tdq(CCPI)              Ind Nam(CSSL)
  Tdq(CCSE)              Ind Nam(CCSO)
  Tdq(CCSO)              Ext           Ena Ope
                             Mod Out Dat(001) Ddn(COUT    )
  Tdq(CDBC)              Ind Nam(CSSL)
  Tdq(CDUL)              Ind Nam(CSSL)
  Tdq(CESE)              Ext           Ena Ope
                             Shr Out Dat(001) Ddn(CEEMSG  )
  Tdq(CESO)              Ext           Ena Ope
                             Shr Out Dat(001) Ddn(CEEOUT  )
  Tdq(CMIG)              Ind Nam(CSSL)  Tdq(CPLI)              Ext           Ena Ope
                             Mod Out Dat(001) Ddn(PLIMSG  )
  Tdq(CRDI)              Ind Nam(CSSL)
+ Tdq(CSCS)              Ind Nam(CSSL)
```

Figure 146. Example CEMT INQUIRE TDQUEUE screen

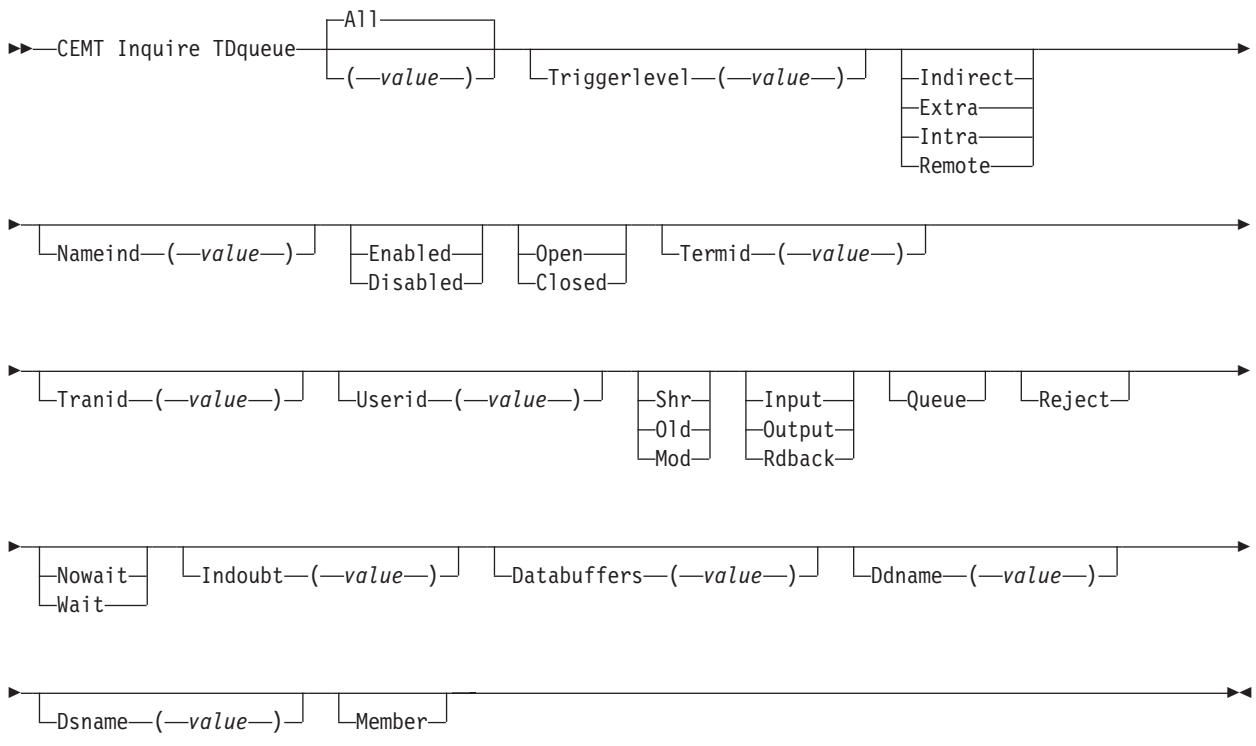
Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 147 on page 297.

```
IN TDQ
RESULT - OVERTYPE TO MODIFY
  Tdqueue(CCSO)
  Type(Extra)
  Nameind()
  Triggerlevel(       )
  Enablestatus( Enabled )
  Openstatus( Open )
  Termid()
  Tranid()
  Userid()
  Disposition(Mod)
  Iotype(Output)
  Indoubt()
  Indoubtwait()
  Databuffers(001)
  Ddname(COUT)
  Dsname(BELL.CICSHT61.JOB04382.D0000105.?)
```

Figure 147. The expanded display of an individual entry

CEMT INQUIRE TDQUEUE



Displayed fields

Databuffers (*value) (extrapartition queues only***)**
displays the number of buffers that can be used by the associated QSAM data set in the range 1 through 255.

Note: An extrapartition queue can be input or output, but not both.

Ddname (*value) (extrapar tition queues only***)**
displays a 1-8 character value that may refer to a data set defined in the startup JCL.

Disposition (*extrapartition queues only***)**
displays the status of the associated data set:

Shr The data set existed before this job step and can be read by other concurrent jobs.

Old The data set existed before this job step.

Mod The system first assumes that the data set exists. For an existing sequential 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.

Dsname (*value) (extrapartition queues only***)**
displays the name of the associated QSAM data set (1-44 characters in length), or a dummy data set (or dummy data set name) that indicates whether the data set will be used by this transient data queue. This option is applicable for output resources only.

Enablestatus (all except indirect and remote queues)
displays a value indicating whether the queue can be accessed by applications. The values are:

Enabled
The queue can be accessed by applications.

Disabled
The queue cannot be accessed by applications, although it can still be open. Queues with names beginning with “C” cannot be disabled because they are usually reserved for use by CICS.

Note:

1. You can reset this value by overtyping it with a different value.
2. A queue can be disabled only if there are no current users. If units of work (UOWs) are currently using the queue, the queue enters a “disable pending” state. The last UOW to complete its operation on the queue fully disables the queue.
3. When a queue is in a “disable pending” state, the value of the ENABLESTATUS option cannot be altered.

Indoubt (*value) (intrapartition queues only***)**
indicates the action CICS is to take for an in-doubt unit of work (UOW) if the definition for this queue specifies WAIT(YES).

Indoubtwait *(intrapartition queues only***)**
specifies whether an in-doubt unit of work (UOW) (which has modified a logically recoverable queue) should wait for resynchronization with its coordinator to determine whether to commit or back out the changes. The values are:

Nowait
The UOW is not to wait. 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 attribute.

This attribute overrides the WAIT parameter defined on the UOW's transaction definition.

Iotype (*extrapartition queues only***)**
displays the type of data set to be used:

Input An input data set.

Output
An output data set.

Rdback
An input data set that is to be read backward.

Member (extrapartition queues only)
returns a 1- to 8-character name that indicates the member name of a partitioned data set. This is blank if the QSAM data set is not a partitioned data set.

Openstatus (extrapartition queues only)
displays whether the extrapartition queue is open or closed. The values are:

OPen The queue is open.

Closed
The queue is closed.

Note:

1. You can reset this value by overtyping it with a different value.
2. Queues no longer have to be enabled before they can be opened or closed. It is advisable for a queue to be disabled before it is opened or closed.

Nameind (*value***) (indirect queues only)**
displays a 4-character string identifying the name of the queue pointed to by the indirect queue.

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 indoubt UOW has issued READQ or DELETEQ requests
- WRITEQ, if the indoubt UOW has issued WRITEQ or DELETEQ requests
- DELETEQ, if the indoubt UOW has issued READQ, WRITEQ or DELETEQ requests.

Reject
The UOW is in-doubt and is waiting. 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 the UOW has made are committed or backed out. Until then, any further requests that need one of the retained locks are rejected, and a LOCKED response is returned. INDOUBT=REJECT causes a LOCKED response to be raised in exactly the same circumstances as those in which INDOUBT=QUEUE causes a transaction to wait.

Tdqueue (*value*)
indicates that this panel relates to a TDQUEUE inquiry and displays the 4-character identifier of a transient data queue. Queue names beginning with "C" are normally reserved for use by CICS.

Termid (*value*)
displays the 4-character name of the terminal or session to be associated with this queue when automatic transaction initiation (ATI) occurs. See also Tranid and Triggerlevel.

Tranid (*value*)
displays the 4-character identifier of the transaction that is to be initiated automatically when the queue trigger level is reached.

Triggerlevel (*value*) (intrapartition queues only)
displays the number of requests for output to a queue that must accrue before automatic transaction initiation (ATI) occurs.

Note: You can reset this value by overtyping it with a different value. The number can be between 0 and 32767.

Type
displays the type of this transient data queue. The values are:
Indirect
The queue type is indirect. The name of the final target queue is shown in the next field.
Extra The queue type is extrapartition.
Intra The queue type is intrapartition.
Remote
The queue type is remote.

Userid
displays the 8-byte identifier associated with the intrapartition queue intended for ATI.

CEMT INQUIRE TERMINAL

Retrieve information about terminals.

Description

INQUIRE TERMINAL returns information about a named terminal defined in the terminal control table (TCT).

In an SNA environment, a terminal is the CICS representation of an SNA logical unit with which CICS can be in communication. (SNA environment means that CICS is using VTAM as the access method for communication with SNA logical units.)

This logical unit can be:

- A physical terminal (such as a 3277)
- A function of a cluster controller (such as a 3790 Type 2 batch function)
- An intersystem communication (ISC) or interregion communication (IRC) session that has been generated by the CEDA transaction
- A remote terminal
- A remote session; that is, a session of a remote connection.

If you are familiar with network names, you might prefer to use the command that uses these names. See "CEMT INQUIRE NETNAME" on page 241.

The returned terminal entry can also be an MVS console.

The message “TEMP OUT SERVICE” can be returned if recovery is in progress for the terminal. The ”TEMP OUT SERVICE” indicator is reset by refreshing the screen, or by overtyping the “INS” field with “INS” or “OUT”. If “TEMP OUT SERVICE” continues to be displayed, the cause of the indicator must be investigated.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE TERMINAL` (the minimum abbreviation is `CEMT I TE`). You get a display that lists the status.
- Type `CEMT INQUIRE TERMINAL` (`CEMT I TE`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i te i at`, the resulting display will show you the details of only those terminals that are in service and available for use. If you enter `cemt i te rem(cicr)`, the display shows the remote terminals (or remote ISC or IRC sessions) owned by the system that the local CICS knows as CICR.

To change various attributes, you can:

- Overtype your changes on the INQUIRE screen after pressing tab to go to the appropriate field (see “Overtyping a display” on page 128).
- Use the CEMT SET TERMINAL command.

(*value***)**
is a terminal identifier (1-4 characters) as specified in an installed terminal definition.

If the terminal name is, for example, S201, this option is coded thus:

```
CEMT INQUIRE TERMINAL(S201)
```

ALl
is the default, for example if you inquire about terminals, you receive information about all terminals, unless you specify a terminal identifier, a terminal class identifier, or a system identifier.

CLass (*value***)**
is the one or two character suffix of a terminal list table (TLT).

If you do not specify a termid or class identifier, by default you receive information about all the terminals.

Sample screen

```
 IN TERM
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Ter(CBRF)           Pri( 000 ) Pag Ins Ati Tti Rte
     Net(CBRF    )                     Rem(CBR ) Rna(CBRF)
  Ter(CERR)           Pri( 000 ) Pag Ins     Tti Loc
     Con(*ERRCON*    )
  Ter(CTSO)           Pri( 000 ) Pag Ins     Tti Loc
     Con(PKING       )
  Ter(KING)           Pri( 000 ) Pag Ins     Tti Loc
     Con(PRUEKING    )
  Ter(RNAL)           Pri( 000 ) Pag Ins     Tti Loc
     Con(INTERNAL    )
  Ter(S225) Tra(CEMT) Pri( 000 ) Pag Ins Ati Tti Loc
     Net(IGCS225 ) Acq     Tas(0000068)           Nqn(GBIBMIYA.IGCS225 )
  Ter(S280) Tra(CEMT) Pri( 000 ) Pag Ins Ati Tti Loc
     Net(IGCS280 ) Acq     Tas(0000063)           Nqn(GBIBMIYA.IGCS280 )
  Ter(TC12) Tra(CEMT) Pri( 000 ) Pag Ins Ati Tti Loc
     Net(IYCWTC12) Acq     Tas(0000072)           Nqn(GBIBMIYA.IYCWTC12)
  Ter(-AAZ)           Pri( 000 ) Aut Ins Ati Tti Ses
     Net(TMPLATE1) Rel                  Rem(CBPS)
  Ter(-AA0)           Pri( 000 ) Aut Ins Ati Tti Ses
     Net(TMPLATE1) Rel Cre              Rem(CBPS)
+ Ter(-AA1)           Pri( 000 ) Aut Out Ati Tti Ses
     Net(TMPLATE1) Rel Cre              Rem(CBPS)
```

Figure 148. CEMT INQUIRE TERMINAL screen

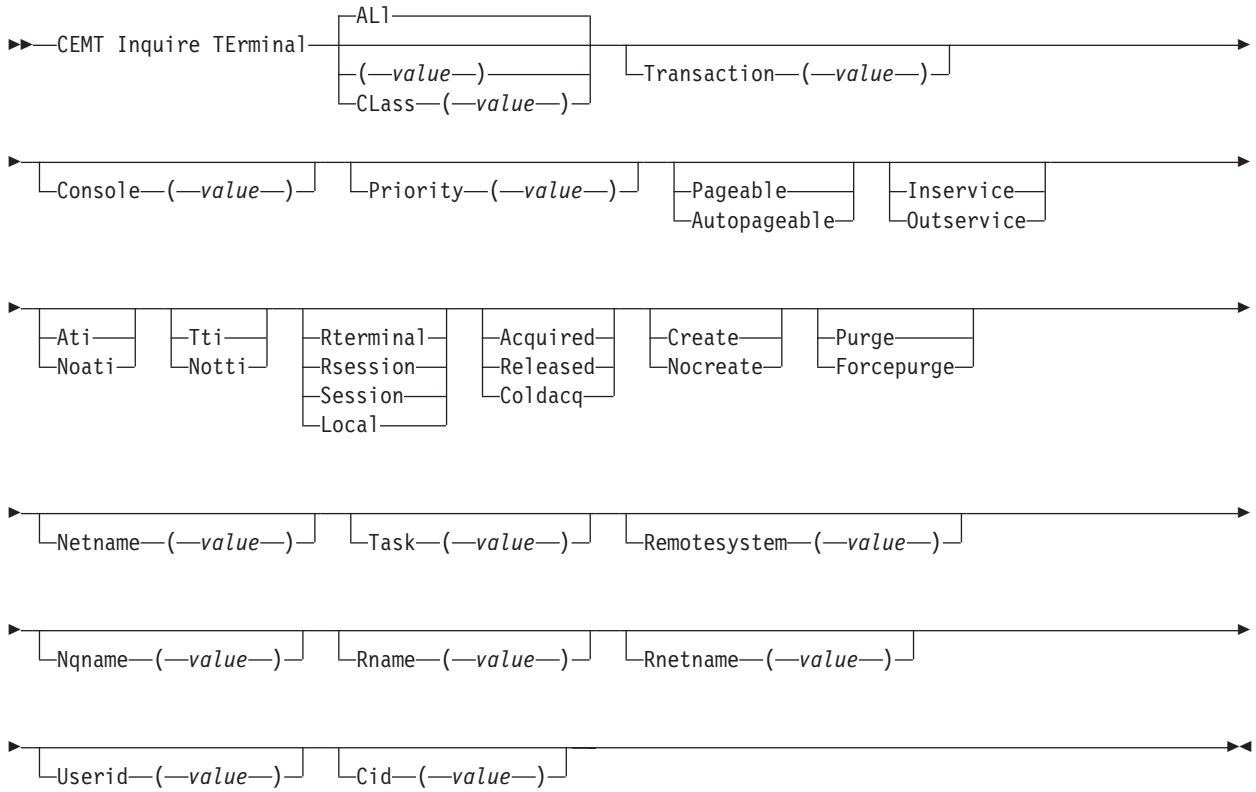
Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by using the tab key (they are shown in the same sequence as in the expanded format), and overtype with input valid for that field. You might find it more convenient to use the expanded format when setting one of these values, (negating is easier because you overtype a displayed field with its negative. If you do thisthe result is not displayed.

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 149.

```
 IN TERM
 RESULT - OVERTYPE TO MODIFY
   Terminal(bell)
   Transaction()
   Priority( 000 )
   Pagestatus( Pageable )
   Servstatus( Inservice )
   Atistatus( Noati )
   Ttistatus( Tti )
   Nature(Local)
   Purgetype(              )
   Netname()
   Console(BELL)
   Termstatus(              )
   Createsess(              )
   Task(00000)
   Remotesystem()
   Nqn(GBIBMIYA.IYCQTC03)
   Rname()
   Rnetname()
+  Cid()
```

Figure 149. The expanded display of an individual entry

CEMT INQUIRE TERMINAL



Displayed fields

AIDCOUNT (*data-area***)**
Returns a fullword binary field giving the number of Automatic Initiate Descriptors (AIDs) queued for the specified terminal. If there are no AIDs, then an AIDCOUNT value of 0 is returned. The count represents the number of AIDs on the CICS system that issued the command.

Atistatus
displays whether the terminal is available for use 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. The values are:
Ati The terminal is available for use.
Noati The terminal is not available for use.

Note:

1. You can reset this value by overtyping it with a different value.
2. A terminal cannot be defined with both NOATI and NOTTI.

Cid (*value***)**
displays an 8-character correlation-identifier that is set only for sessions, and only after a session has been acquired. (The value relates to the previously acquired session if there was one, if the session is not acquired). CID is set as follows:

- For LU6.2 sessions, it is an 8-character token that is common to the two end sessions that are connected together.
- For MRO sessions, it is set to the termid of the session on the system to which this session is connected.
- For LU6.1 sessions to IMS, it is set to the value of NETNAMEQ as defined or negotiated.
- For other LU6.1 sessions, it is set to the value supplied by the other end of the 6.1 session.

Using CID, you can relate the two parts of an LU6.2, MRO, or LU61 conversation together:

- For LU6.2, you can inquire on the same CID.
- For MRO, you can locate the session with the terminal name.
- For LU6.1 to IMS, you can inquire on the netname.
- For other LU6.1, you can use the name to inquire on the connected system.

Console (*value***)**
displays, for an MVS console only, a 12-byte string that contains the identifier of the console, in two parts. 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.

If the console is defined by a numeric identifier, the string is divided into two parts, separated by a period (.) in the ninth byte position. The 12-byte string contains the following information:

- The first 8 bytes contain the MVS console name, if it is known, or the string '*UNKNOWN' if the name is not known.
- A period separating the name from the numeric ID.
- The last 3 bytes contain the numeric console ID.

Createsess (*vtam only***)**
indicates whether or not the terminal can be acquired automatically by ATI transactions. You cannot specify this for IRC sessions. The values are:

Create
If the terminal is not in session, CICS acquires it if required for an ATI request.

Nocreate
If the terminal is not in session, CICS does not acquire it to satisfy an ATI request. A session must be started by, for example, a logon request or a CEMT SET TERMINAL ACQUIRED command before the ATI request can be satisfied.

If NOCREATE is set for an LU6.1 ISC session and there are no allocatable sessions left, the connection is placed OUTSERVICE.

You can rest this value by overtyping it with a different value.

Nature
indicates the nature of the terminal (whether it is a physical device or a session; and whether it is local or remote). The values are:

Rterminal
The terminal is a remote device.

Rsession
The terminal is a session of a remote connection.

Session
The terminal is an ISC or IRC session.

Local The terminal is directly attached to this system and is a physical device or a function of a cluster controller.

Netname (*value***)**
displays an 8-character network name.

For a physical terminal, the Netname is the name by which this terminal is known to VTAM.

For ISC sessions, it is the name by which the session (or group of sessions) is known to VTAM.

For IRC sessions, it is the name used by the connected region to log on to the interregion communication program (DFHIRP).

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 TS for z/OS, Version 3.2 region, Netname is blank.)

Note: If the netname is a VTAM LU alias, it is different from the netname component of the NQNAME, which always contains the real netname.

Nqname (*value***)**
displays the 17-character network-qualified name.

Remote terminals do not have an NQNAME value.

If the status (Termstatus) is RELEASED, NQNAME displays the netname.

Pagestatus
displays whether pages after the first in a series are written to the terminal on request from the operator or automatically. The values are:
Pageable
Pages are written on request.
Autopageable
Pages are written automatically.

Note: You can reset this value by overtyping it with a different value.

Priority (*value***)**
displays a 3-character string identifying the priority of a terminal relative to other terminals. The priority of a task is the sum of the transaction priority, the terminal priority, and the operator priority. Priority has no meaning for terminals that are ISC sessions being used as alternative facilities.

Note: You can reset this value by overtyping it with a different value.

The value is in the range 0–255, where 255 is the highest priority.

Purgetype (*input only field***)**
specifies whether transactions running with the named terminal are to be purged. The values are:
Purge
Transactions is terminated only if system and data integrity can be maintained. A transaction is purged if its definition specifies SPURGE=NO.
Forcepurge
Transactions are to be purged immediately. This can lead to unpredictable results and is for use in exceptional circumstances.

Remotesystem (*value***)**
displays 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 toward the terminal-owning region (TOR), or an indirect connection that provides the netname of the TOR.

Otherwise this field is blank.

Rname (*value***)**
displays the 4-character name by which this terminal is known in the system in which it is defined (the TOR). RNAME applies only to terminals defined as remote; for others, blanks are displayed.

Rnetname (*value***)**
displays the 8-character netname of the owning TOR, if the subject of the inquiry is a remote terminal. If this field is blank **and** the terminal is remote, the indirect connection, displayed in REMOTESYSTEM, contains the netname of the TOR.

Servstatus
displays whether the terminal is available for use. The values are:

Inservice
The terminal is available for use. For VTAM, Ins(ervice) means that the terminal can be ACQUIRED. For IRC sessions, Ins(service) means that the connection to the MRO partner is INSERVICE.

Outservice
The terminal is not available for use. Setting a terminal Out(service) means that the terminal can no longer be used by transactions. If PURGE or FORCEPURGE is also specified, any transaction using the terminal is terminated abnormally. If PURGE or FORCEPURGE is not specified, the transaction is allowed to terminate normally, but no further transactions are allowed to use the terminal. For VTAM, setting a terminal Out(service) also causes it to be released and the operator to be signed off, either immediately or when the current transaction has terminated. For IRC sessions, Out(service) means that the connection to the MRO partner is OUTSERVICE.

In an LU6.1 ISC session, the connection is set Out(service) if there are no allocatable sessions left.

Note: For physical terminals and LU6.1 sessions, you can reset this value by overtyping it with a different value.

Task (*value***)**
displays the task number of the transaction that is running on this terminal.

Terminal (*value***)**
indicates that this panel relates to a TERMINAL inquiry and displays a 4-character terminal identifier as defined in an installed terminal definition. This includes all terminals and sessions, but not logical device codes (LDCs), model TCTTEs, mode groups, or system entries. See also **Net**name.

Termstatus (*vtam only***)**
displays whether CICS is in session with the logical unit represented by this terminal. The values are:

Acquired
CICS is in session with the logical unit represented by the terminal.

Released
CICS is not in session with the logical unit represented by the terminal. If you set this option to RELEASED, a session is terminated immediately if you also specify the PURGE option, otherwise the session is terminated when the current active transaction finishes.

For physical terminals and LU6.1 sessions, you can reset this value by overtyping it with a different value or with:

Coldacq
CICS is in session with the logical unit represented by the terminal, where no resynchronization is required.

Transaction (*value***)**
displays a 4-character string identifying the name of the transaction currently being processed with this terminal as its principal facility or as a secondary facility.

Ttistatus
displays whether the terminal can be used by the transactions that are initiated from this terminal. The values are:

Tti This terminal can be used by transactions.
Notti This terminal cannot be used by transactions.

Note:

1. You can reset this value by overtyping it with a different value.
2. A terminal cannot be defined with both NOATI and NOTTI.

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 userid—as specified in the DFLTUSER system initialization parameter—is returned.

CEMT INQUIRE TRANSACTION

Retrieve information about transactions.

Description

INQUIRE TRANSACTION returns information about transaction definitions.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE TRANSACTION` (the minimum abbreviation is `CEMT I TRANS`). You get a display that lists the current status.
- Type `CEMT INQUIRE TRANSACTION` (`CEMT I TRANS`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i trans en pu`, the resulting display will show you the details of only those transactions that are enabled and system-purgeable.

To change various attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 128).
- Use the CEMT SET TRANSACTION command.

(*value***)**
is a 1–4 character transaction identifier. Only transactions that have been defined in the CICS system definition (CSD) file and installed on the running CICS system are accessible through CEMT.

ALl
is the default.

CLass(*value*)
is the 2-character suffix of a transaction list table (XLT).

Sample screen

```
 IN TRAN
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Tra(AADD) Pri( 001 ) Pro(DFH$AALL) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
  Tra(ABRW) Pri( 001 ) Pro(DFH$ABRW) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
  Tra(ADYN) Pri( 001 ) Pro(DFH99   ) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
  Tra(AINQ) Pri( 001 ) Pro(DFH$AALL) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
  Tra(AMNU) Pri( 001 ) Pro(DFH$AMNU) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
  Tra(AORD) Pri( 001 ) Pro(DFH$AREN) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
  Tra(AORQ) Pri( 001 ) Pro(DFH$ACOM) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
  Tra(AREP) Pri( 001 ) Pro(DFH$AREP) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
+ Tra(AUPD) Pri( 001 ) Pro(DFH$AALL) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
```

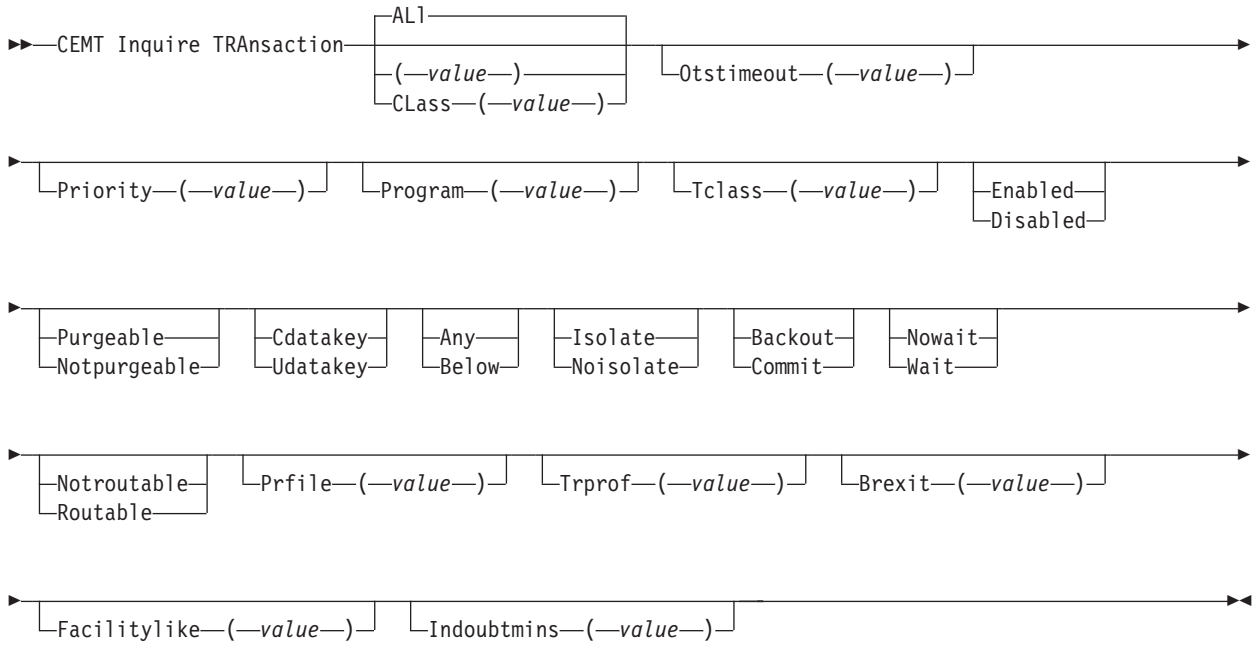
Figure 150. CEMT INQUIRE TRANSACTION screen

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 151.

```
IN TRAN
RESULT - OVERTYPE TO MODIFY
  Transaction(AADD)
  Priority( 001 )
  Program(DFH$AALL)
  Tclass( DFHTCL00 )
  Status( Enabled )
  Routing(Dynamic)
  Routestatus(Notroutable)
  Purgeability( Purgeable )
  Prfile(DFHCICST)
  Taskdatakey(Udatakey)
  Taskdataloc(Below)
  Isolatest(Isolate)
  Trprof()
  Indoubt
  INdoubtwait
  Brexit
  Facilitylike
  Otstimeout(12000)
```

Figure 151. The expanded display of an individual entry

CEMT INQUIRE TRANSACTION



Displayed fields

Brexit(*value***)**
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.

Facilitylike(*value***)**
returns the 4-character name of the terminal defined by the FACILITYLIKE parameter of the PROFILE associated with the named transaction resource definition.

Indoubt
displays the action (based on the ACTION attribute of the TRANSACTION resource definition) 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 normally taken until the time returned in Indoubtmins expires. (For exceptions to this rule, see Indoubtwait.)

The 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.

Indoubtmins(*value***)**
displays 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.

See also Indoubt and Indoubtwait.

Indoubtwait
displays (based on the WAIT attribute of the TRANSACTION definition) how CICS is to respond if a failure occurs while a unit of work (UOW) is in an in-doubt state. The 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.

For further information about the meaning of the ACTION and WAIT attributes of the TRANSACTION definition, see the *CICS Resource Definition Guide*.

Isolatest
displays whether the transaction's user-key task-lifetime storage is isolated from the user-key programs of other transactions. The values are:

Isolate
The transaction's user-key task-lifetime storage is accessible only by the user-key programs of its own task. The transaction's user-key task-lifetime storage is isolated from all the user-key programs of all other tasks.

Noisolate
The transaction's user-key task-lifetime storage is accessible by its own programs, and also by user-key programs of other transactions defined with the ISOLATE(NO) option.

Otstimeout(*value*)
displays a fullword data-area containing the default period in seconds that an OTS transaction created in an EJB environment executing under this CICS transaction, will be allowed to execute prior to syncpoint.

Prfile(*value*)
displays the name of the profile definition that defines additional options associated with this transaction.

Priority(*value*)
displays a value indicating the priority of a transaction relative to other transactions. When a transaction is running as a CICS task, the priority of a task is the sum of the transaction priority, the terminal priority, and the operator priority.

Note: You can reset this value by overtyping it with a different value. The value is in the range 0–255, where 255 is the highest priority.

Program(*value*)
displays an 8-character string identifying the name of the first program to be executed when this transaction is started.

Purgeability
displays whether the transaction is purgeable in system stall conditions. The values are:

Purgeable
The transaction is system-purgeable. This value relates to the SPURGE parameter on the transaction resource definition and indicates that CICS can purge the transaction in a deadlock time-out situation. See the *CICS Resource Definition Guide* for information about the SPURGE and DTIMEOUT parameters on a transaction resource definition.

Notpurgeable
The transaction cannot be purged.

Note: You can reset this value by overtyping it with a different value.

Routstatus
displays whether, if the transaction is the subject of an eligible EXEC CICS START command, it will be routed using the enhanced routing method. The values are:

Notroutable
If the transaction is the subject of a START command, it will be 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*.

Status
displays whether the transaction is available for use. The values are:

Enabled
The transaction is available for use.

Disabled
The transaction is not available for use.

Note: If a transaction is disabled, this does not prevent a START command that names this transaction from being shipped to a remote region. When a task is attached for the requested transaction, CICS checks that the transaction is enabled in the remote region.

Note: You can reset this value by overtyping it with a different value.

Taskdatakey
displays the storage key in which CICS obtains all storage for use by the transaction. 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 the transaction.

The values are:

Cdatakey
CICS obtains storage for the transaction from CICS-key storage.

Application programs that execute in CICS key have read-write access to this storage, but user-key programs have read-only access.

Udatakey
CICS obtains storage for the transaction from user-key storage. Application programs that execute in any key have read-write access to this storage.

See the description of the TASKDATAKEY parameter on the transaction resource definition in the *CICS Resource Definition Guide*.

Taskdataloc
displays whether certain CICS control blocks (including EIB and TWA) for a transaction are acquired above or below the 16MB line. The values are:

Any The transaction accepts task-related data anywhere.

Below
The transaction requires any task-related data (TWA and EIB plus any internal control blocks) to be located below the 16MB line.

Tclass(*value***)**
displays an 8-character string identifying the name of the transaction class to which the transaction belongs. If the transaction does not belong to a class, DFHTCL00 is returned.

Note: You can reset this value by overtyping it with a different value.

To remove a transaction from its TCLASS, set this field to DFHTCL00. An added or changed TCLASS **must** be one that has already been defined.

Transaction(*value***)**
indicates that this panel relates to a TRANSACTION inquiry and displays a 4-character transaction identifier. Only transactions that have been defined in the CICS system definition (CSD) file and installed on the running CICS system are accessible through CEMT.

Trprof(*value***)**
displays the name of the transaction routing profile that defines additional options associated with this transaction if it is defined as a remote transaction.

CEMT INQUIRE TRDUMPCODE

Retrieve information about transaction dump codes.

Description

The INQUIRE function allows you to see the current settings of the entries in the transaction dump table. See the *CICS Problem Determination Guide* for information on using dumps.

Note: SHUTDOWN and MAXIMUM must not be used in the same command.

This command returns entries that were added to the dump table by a user application program or CEMT command, and also entries that CICS added to the dump table through dump requests. Be aware that the entries that CICS added to the dump table are temporary, and they are not restored during a warm start, or during a cold or initial start. Entries added by a user application program or CEMT command are restored during a warm start, but not during a cold or initial start.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE TRDUMPCODE` (the minimum abbreviation is `CEMT I TRD`). You get a display that lists the current status.
- Type `CEMT INQUIRE TRDUMPCODE` (`CEMT I TRD`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i trd sy sh`, the resulting display will show you the details of only those transaction dump codes for which system dumps are to be taken and for which the CICS system is to shut down.

To change various attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see "Overtyping a display" on page 128).
- Use the CEMT SET TRDUMPCODE command.

(*value***)**
is a 4-byte transaction dump code. See *CICS Messages and Codes* for information on transaction dump codes.

ALl
is the default.

Sample screen

```
IN TRDU
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Trd(ATNI) Tra         Loc Max( 999 ) Cur(0002)
 Trd(ERRS) Tra         Loc Max( 999 ) Cur(0001)
```

Figure 152. CEMT INQUIRE TRDUMPCODE screen

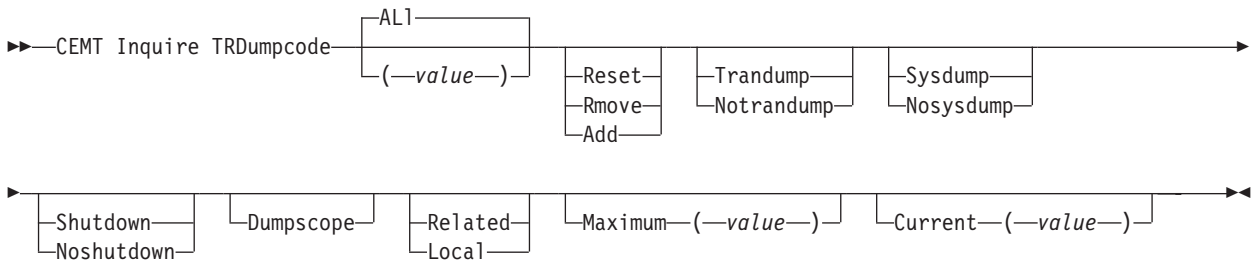
Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 153.

```
IN TRDU
RESULT - OVERTYPE TO MODIFY
  Trdumpcode(ATNI)
  Trandumping( Trandump )
  Sysdumping( Nosysdump )
  Shutoption( Noshutdown )
  Dumpscope( Local )
  Maximum( 999 )
  Current(0002)
  Action(              )
```

Figure 153. The expanded display of an individual entry

CEMT INQUIRE TRDUMPCODE



Displayed fields

Action (*input only field***)**
displays whether the specified transaction dump code is to be reset, removed from the dump code table, or added to the dump code table. The values are:

Reset
The current count of dump calls for this dump code is to be reset to zero.

Remove
This code is to be removed from the dump code table.

Add The specified transaction dump code is to be added to the dump code table.

Current(*value***)**
displays the number of dump calls that have been made for this transaction dump code since the number was last reset to zero. The CURRENT value can be reset explicitly (using a SET SYDUMPCODE RESET command), or automatically whenever CICS shuts down.

Note: You can reset this value by overtyping it with a different value.

Dumpscope
displays whether a system dump request is to be sent to MVS images in the sysplex which are running XCF/MRO connected CICS regions related to the CICS region on which the dump is initiated.

Note: A related CICS region is one in which the unit of work identifiers, in the form of APPC tokens, of one or more tasks match those in the CICS region that issued the dump request.

Related
A system dump request is to be sent for this system dump code.

Local A system dump request is not to be sent to other MVS images in the sysplex for this system dump code.

Note: You can reset this value by overtyping it with a different value.

Maximum(*value***)**
displays the largest number of dump calls for this dump code that result in

dumps being taken. The value is in the range 0-999 and can be specified by using TRDUMAX in the SIT parameter. A value of 999 means the default, 'no limit'.

Shutoption
displays whether the CICS system is to shut down after the occurrence of an error corresponding to this dump code. The values are:

Shutdown
The CICS system is to shut down.

Noshutdown
The CICS system is not to shut down.

Note: You can reset this value by overtyping it with a different value.

Sysdumping
displays whether a system dump is to be taken for this transaction dump code. The values are:

Sysdump
A system dump is to be taken.

Nosysdump
A system dump is not to be taken.

Note: You can reset this value by overtyping it with a different value.

Trandumping
displays whether a transaction dump is to be taken for this transaction dump code. The values are:

Trandump
A transaction dump is to be taken.

Notrandump
A transaction dump is not to be taken.

Note: You can reset this value by overtyping it with a different value.

Trdumpcode(*value*)
indicates that this panel relates to a TRDUMPCODE inquiry and displays a 4-character transaction dump code. See *CICS Messages and Codes* for information on transaction dump codes.

CEMT INQUIRE TSMODEL

Retrieve information about TS models.

Description

The INQUIRE TSMODEL command returns information about temporary storage models (TS models).

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE TSMODEL` (the minimum abbreviation is `CEMT I TSM`). You get a display that lists the current status.

- Type CEMT INQUIRE TSMODEL (CEMT I TSM) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter cemt i tsm main, the resulting display will show you the details of only those temporary storage models associated with temporary storage queues which are to be resident in main storage.

To change various attributes, overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 128).

ALl
is the default.

(*value***)**
is the name of the temporary storage model for which information is requested.

Sample screen

```
I TSM
STATUS:  RESULTS
 Tsm(DFHWEB  ) Pre(DFHWEB          ) Mai
```

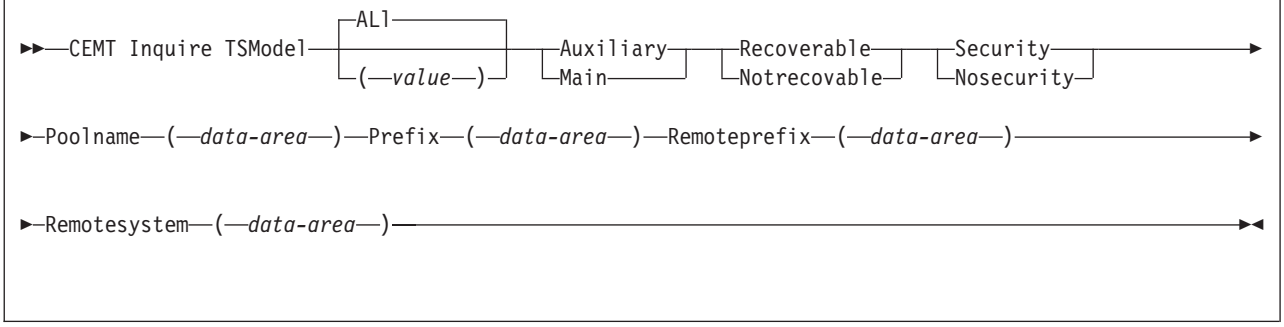
Figure 154. CEMT INQUIRE TSMODEL screen

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 155.

```
 I TSM
 RESULT
   Tsmodel(DFHWEB)
   Prefix(DFHWEB)
   Location(Main)
   Recovstatus(Notrecovable)
   Securityst(Nosecurity)
   Poolname()
   Remotesystem()
   Remoteprefix()
```

Figure 155. The expanded display of an individual entry

CEMT INQUIRE TSMODEL



Displayed fields

Location
displays the location where temporary storage queues associated with this TS model reside. The values are:

Auxiliary
TS queues associated with this TS model are held on the CICS temporary storage VSAM data set DFHTEMP, or in temporary storage pools in the coupling facility.

Main TS queues associated with this TS model are held in main storage.

Poolname(*value*)
displays an 8-character shared pool name.

Prefix(*value*)
displays the prefix for this model as a 16-character string.

Recovstatus
displays the recovery status of the temporary storage queue. The values are:
Recoverable
The temporary storage queue is recoverable.
Notrecovable
The temporary storage queue is not recoverable.

Remoteprefix(*value*)
displays the remoteprefix for this model as a 16-character string.

Remotesystem(*value*)
displays the remotesystem name for this model as a 4-character string.

Securityst
displays the security status for this model. The values are:

Security
security checking will be performed for queue names matching this model.

Nosecurity
security checking will not be performed for queue names matching this model.

Tsmodel(*value*)
indicates that this panel relates to a TSMODEL inquiry and displays the 8-character name of a temporary storage model.

Note: Nondisplayable characters appear as periods. You can use PF2 on the expanded panel to see the value in hexadecimal.

CEMT INQUIRE TSPOOL

Retrieve information about a TS pool.

Description

The INQUIRE TSPOOL command returns information about a temporary storage pool (TS pool).

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE TSPOOL` (the minimum abbreviation is `CEMT I TSP`). You get a display that lists the current status.
- Type `CEMT INQUIRE TSPOOL` (the minimum abbreviation is `CEMT I TSP`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i tsp connected`, the resulting display will show you the details of only those shared temporary storage pools that are connected.

To change various attributes, overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 128).

ALl
is the default.

(*value***)**
is the name of the temporary storage pool for which information is requested.

Sample screen

```
I TSP
STATUS:  RESULTS
 Tsp(DFHWEB  ) Unc
```

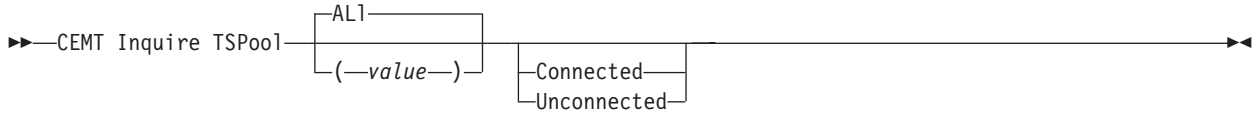
Figure 156. CEMT INQUIRE TSPOOL screen

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 157.

```
I TSP
RESULT
  Tspool(DFHWEB)
  Connstatus(Unconnected)
```

Figure 157. The expanded display of an individual entry

CEMT INQUIRE TSPOOL



Displayed fields

Connstatus
displays the connection status of this TS pool. The values are:
Connected
This TS pool is connected.
Unconnected
This TS pool is not connected.

Tspool(*value***)**
indicates that this panel relates to a TSPOOL inquiry and displays the 8-character name of a temporary storage pool.

Note: Nondisplayable characters appear as periods. You can use PF2 on the expanded panel to see the value in hexadecimal.

CEMT INQUIRE TSQUEUE / TSQNAME

Retrieve information about temporary storage queues. This section applies also to the alternative command, **CEMT INQUIRE TSQNAME**. Use either to inquire about names up to 8 characters long, but use INQUIRE TSQNAME to inquire about names up to 16 characters long.

Description

The INQUIRE TSQUEUE command returns information about temporary storage queues (TS queues). The INQUIRE TSQUEUE command operates on all the temporary storage queues that exist in the CICS region, including those created internally by CICS for use by CICS itself (for example, queues used by BMS). You can identify the temporary storage queues created by CICS for its own use by queue names that begin with the following character strings:
****** BMS paging
$$ BMS route
X'fa' to X'ff'
CICS
CEBR Default CEBR queue name
DF CICS
DFHM Message cache for message-protected tasks
DFxxxx
CICS REQIDS (where x is hexadecimal)

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE TSQUEUE` (the minimum abbreviation is `CEMT I TSQ`). You get a display that lists the current status.

- Type CEMT INQUIRE TSQUEUE (CEMT I TSQ) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter cemt i tsq main, the resulting display will show you the details of only those temporary storage queues that are resident in main storage.

To change various attributes, overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see "Overtyping a display" on page 128).

ALl
is the default.

(*value***)**
is the name of the temporary storage queue for which information is requested.

Sysid(*value***)**
specifies the system name that corresponds to the temporary storage pool name if the request is to be sent to a TS server.

This is an input-only value, and must be specific—you cannot specify generic names for SYSID. If the SYSID is not found in a TST TYPE=SHARED entry, CICS returns a SYSID NOT FOUND error.

Poolname(*value***)**
specifies the 8-character POOLNAME of a temporary storage pool if the request is to be sent to a TS server. Specify Poolname to view shared temporary storage queues.

If the POOLNAME is not found, CICS returns a POOLERR.

Sample screen

```
INQ TSQ
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Tsq(AXBYQUEUENAME1  ) Num(00003) Len(0000000192) Mai
    Tra(CECI)     Max(00064) Min(00064) Las( 00001267 )
 Tsq(RDOQUEUENAME4   ) Num(00001) Len(0000000064) Aux
    Tra(CECI) Rec Max(00064) Min(00064) Las( 00000097 )
 Tsq(RDOQUEUENAME5   ) Num(00002) Len(0000000128) Aux
    Tra(CECI) Rec Max(00064) Min(00064) Las( 00000086 )
 Tsq(RDQUEUENAME5    ) Num(00003) Len(0000000192) Mai
    Tra(CECI)     Max(00064) Min(00064) Las( 00000076 )
```

Figure 158. CEMT INQUIRE TSQUEUE screen

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 159.

```
INQ TSQ
RESULT - OVERTYPE TO MODIFY
  Tsqueue(AXBYQUEUENAME1)
  Numitems(00003)
  Length(0000000192)
  Location(Main)
  Action(             )
  Transid(CECI)
  Recovstatus(Notrecovable)
  Maxitemlen(00064)
  Minitemlen(00064)
  Lastusedint(00001357)
```

Figure 159. The expanded display of an individual entry

If you place the cursor against a specific entry in the list and type a ‘B’ (upper or lower case), CICS invokes a direct link to the program DFHEDFBR used by the CEBR transaction to enable you to browse the contents of this TS queue, (see Chapter 8, “CEBR - temporary storage browse,” on page 75 for details).

Security warning:: Take care to avoid a security exposure, because in this situation, access to the browse function is not ruled by transaction attach security checking using the XTRAN parameter in DFHSIT, as it is for the CEBR transaction, but instead, access is ruled by application program security checking using the XPPT parameter in DFHSIT.

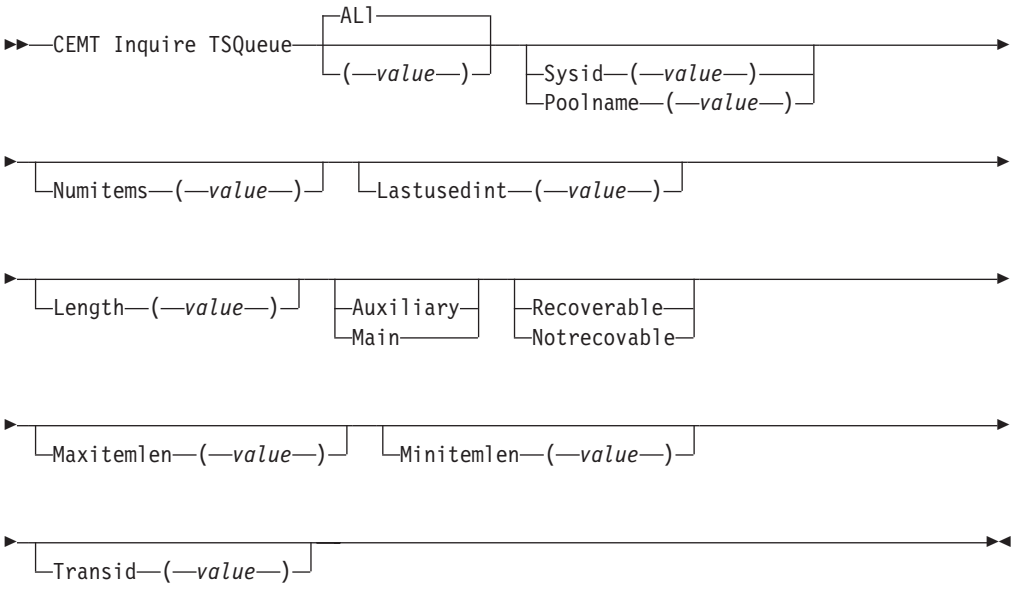
Not all TS queues visible on the CEMT INQUIRE TSQUEUE results panel are necessarily browsable via CEBR. Specifically, this applies to queues resident in a shared TS POOL when the POOLNAME is specified on the INQUIRE TSQUEUE request.

Such queues are browsable if there is a tsmodel or TST entry mapping the queue name to the TSPOOL.

Note: For this release only you can also browse queues in a shared pool by specifying a SYSID which maps to the shared pool via an entry in the SYSID table. However, the SYSID required for the mapping is not available to CEMT, so entering ‘B’ next to the queue at first gives a NOT FOUND response from CEBR. If an appropriate SYSID is then provided by the user, the reference to the shared pool can be resolved.

If there is no suitable entry in the SYSID table and no TSMODEL mapping the queue name to the shared pool, the queue cannot be browsed and CEBR will indicate that the requested queue does not exist.

CEMT INQUIRE TSQUEUE/TSQNAME



Displayed fields

Lastusedint(*value***)**
displays the elapsed time interval in binary seconds since the queue was last referenced.

Length(*value***)**
displays the total length in bytes of all the items 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.

Location
displays where the temporary storage queue resides. The values are:
Auxiliary
The temporary storage queue is held on the CICS temporary storage VSAM data set DFHTEMP, or in temporary storage pools in the coupling facility.
Main The temporary storage queue is held in main storage.

Maxitemlen(*value***)**
displays 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.

Minitemlen(*value***)**
displays the length in bytes of the smallest item in the temporary storage queue.

The length of a queue item is the sum of the length of the user data plus 8 bytes for header information, rounded up. For main 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.

Numitems(*value*)
displays the number of items in the temporary storage queue.

Recovstatus
displays the recovery status of the temporary storage queue. The values are:
Recoverable
The temporary storage queue is recoverable.
Notrecovable
The temporary storage queue is not recoverable.

Transid(*value*)
displays the 4 character id of the transaction that created the TS queue.

TSQueue(*value*)
indicates that this panel relates to a TSQUEUE inquiry and displays the 16-character name of a temporary storage queue.

Note: Nondisplayable characters appear as periods. You can use PF2 on the expanded panel to see the value in hexadecimal.

CEMT INQUIRE UOW

Retrieve information about units of work (UOWs).

Description

INQUIRE UOW returns information about a named unit of work, or about all the UOWs currently in the system. It displays the state of the UOW (for example, INDOUBT) and whether it is active, waiting, or shunted.

If you suspect a problem with either a recoverable data set or a connection, you can use INQUIRE UOW to display UOWs that have been shunted due to a connection or data set failure. The command, in some cases, displays 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*.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type CEMT INQUIRE UOW (the minimum abbreviation is CEMT I UOW). You get a display that lists all the current UOWs.
- Type CEMT INQUIRE UOW (CEMT I UOW) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter cemt i uow sh con, the resulting display will show you the details of only those UOWs that have been shunted due to the failure of a connection.

(*value***)**
is a 16-character local identifier of a UOW.

ALl
is the default. If you do not specify a UOW identifier, you receive information about all UOWs in the system.

Sample screen

```
IN UOW
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Uow(AB8FFF37584B6601) Inf Act Tra(CSSY) Tas(0000005)
    Age(00004090)                         Use(BELL    )
 Uow(AB8FFF3758786C01) Inf Act Tra(CSSY) Tas(0000006)
    Age(00004090)                         Use(BELL    )
 Uow(AB8FFF376D9B6601) Inf Act Tra(CSTP) Tas(0000008)
    Age(00004090)                         Use(BELL    )
 Uow(AB900BD865417C04) Inf Act Tra(CSNE) Tas(0000018)
    Age(00000700)                         Use(BELL    )
 Uow(AB900BE2010AC401) Inf Act Tra(CEMT) Tas(0000043)
    Age(00000689) Ter(S21D) Netn(IGBS21D ) Use(BELL    )
```

Figure 160. CEMT INQUIRE UOW screen

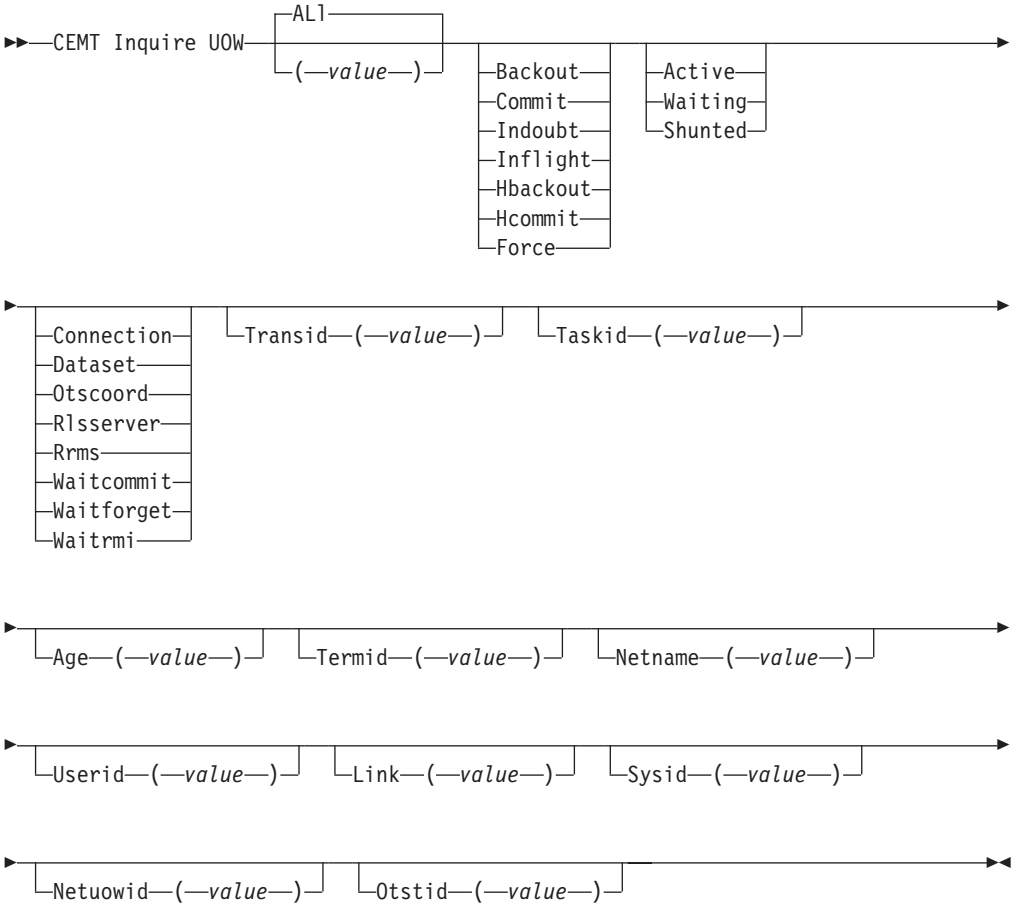
Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 161.

```
IN UOW
RESULT - OVERTYPE TO MODIFY
  Uow(AB8FFF37584B6601)
  Uowstate( Inflight )
  Waitstate(Active)
  Transid(CSSY)
  Taskid(0000005)
  Age(00004233)
  Termid()
  Netname()
  Userid(BELL)
  Waitcause()
  Link()
  Sysid()
  Netuowid(..GBIBMIYA.CICSHT61........)
  Otstid()
```

Figure 161. The expanded display of an individual entry

CEMT INQUIRE UOW



Displayed fields

Age (*value*)
displays the age of the UOW in seconds. This is the number of seconds since the UOW entered its current state, or since the start of the UOW.

Link(*value*)
displays the 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 displays blanks.

Netname (*value*)
displays 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 displays the network name of the remote region.

If the UOW was not started from a terminal, nor from an ISC or MRO session, NETNAME displays blanks.

Netuowid (*value*)
displays a 27-character string containing the LU6.2 name for the UOW within this network—that is, the network-wide identifier of the UOW.

Note: Nondisplayable characters appear as periods. You can use PF2 on the expanded panel: to see the value in hexadecimal.

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(*value*)
displays the first 128 bytes of the transaction identifier (TID) of the OTS transaction of 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 (*value*)
displays the 4-character sysid that, for a WAITCAUSE value of CONNECTION, is the sysid of the connection that caused the UOW to wait ir be shunted. If the connection has been discarded, and for other WAITCAUSE values (including OTSCOORD), SYSID returns blanks.

Taskid (*value*)
displays 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 (*value*)
displays 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 OUWs that are part of an OTS transaction, it will be the session used by the request stream that attached the task.

Transid (*value*)
displays the 4-character ID of the transaction that started this UOW.

UOW(*value*)
indicates that this panel relates to a UOW inquiry and displays a 16-character UOW local identifier.

Uowstate
displays the state of the UOW. The 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.

Indoubt
This UOW is in the in-doubt state.

Inflight
The UOW is running normally.

Hbackout
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 have been backed out with a CEMT SET UOW command.

Hcommit
The UOW has been forcibly committed.

Note: You can reset this value by overtyping it with:

Commit
Attempt to force syncpoint commit processing, as specified for this UOW.

Backout
Attempt to force syncpoint backout processing, as specified for this UOW.

Force
Attempt to force the UOW to backout or commit, as specified on the ACTION option of the TRANSACTION resource definition.

All these values are valid only for UOWs that have been shunted in-doubt. For information about the in-doubt attributes of TRANSACTION resource definition, see the *CICS Resource Definition Guide*.

Userid(*value***)**
displays the 8-character user ID with which this transaction was started.

Waitcause
displays, if the UOW is waiting or shunted, the type of resource that caused it to be so.

Note: In the case of a wait, it is the UOW that is waiting, not the task.

The values are:

Connection
This UOW is waiting or was shunted due to 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 was shunted due to the failure of 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 was shunted due to the failure of the OTS coordinator.

Rlsserver
This UOW is waiting or was shunted for the OTS coordinator due to the failure of an RLS server.

Rrms This UOW is waiting or has been shunted because communication has been lost with RRS/MVS.

Waitcommit
This UOW is waiting or was 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
displays whether syncpoint processing of the UOW has been deferred. The values are:

Active
The UOW is running normally, or is being unshunted.

Waiting
Syncpoint processing has completed on this system, but not on all systems involved in the distributed UOW. Values of WAITFORGET or WAITRMI, and BACKOUT or COMMIT indicate how the UOW was resolved on this system.

Shunted
Syncpoint processing of the UOW has been deferred. SHUNTED further indicates that the task, terminal and program storage have been released, and locks have been retained.

CEMT INQUIRE UOWDSNFAIL

Obtain information about failed units of work that have updated CICS file-control-managed data sets.

Description

INQUIRE UOWDSNFAIL returns information about UOW failures that are associated with specific data sets. If you enter the basic command (CEMT INQUIRE UOWDSNFAIL) without any attributes to act as a filter, CICS displays all failed UOWs for all data sets.

Because this command returns information about UOW failures, if a UOW that had suffered a failure is in the process of being retried when the command is issued, the UOW will not be displayed by INQUIRE UOWDSNFAIL (it is not currently shunted or failed). If the retry is successful, the UOW no longer has a failure with respect to that particular data set. If the retry is not successful, the UOW will be shunted again, and INQUIRE UOWDSNFAIL will display it as failed with respect to the data set.

Alternatively, if you are interested only in those failed UOWs for specific data sets, you can use the DSNAME parameter as a filter by specifying either a specific or generic data set name. You can also use other attributes as filters, such as CONNECTION for UOWs that failed indoubt, or DATASET for UOWs that failed during backout for some reason. See the list of reasons for UOW failures, and the associated causes, which you can use for filtering the results of INQUIRE UOWDSNFAIL commands.

Input

Press the Clear key to clear the screen and enter the basic command as `CEMT INQUIRE UOWDSNFAIL`. (The minimum abbreviation is `CEMT I UOWD`.) You get a display of all the failed UOWs, if any. Otherwise, CICS returns the NOT FOUND error condition

Alternatively, enter the command with one or more valid attributes to filter the results of the inquiry. For example, if you enter `cemt i uowd ds(applb.*)`, the resulting display will show you the details of only those failed UOWs that are associated with data sets with a high-level qualifier of APPLB.

Sample screen

```
INQ UOWDSN
STATUS:  RESULTS
 Dsn(RLS.ACCOUNTS.ESDS.DBASE1                  ) Dat Del
    Uow(AA6DB080C40CEE01)                       Rls
 Dsn(RLS.ACCOUNTS.ESDS.DBASE1                  ) Dat Ind
    Uow(AA6DB08AC66B4000)                       Rls
```

Figure 162. CEMT INQUIRE UOWDSNFAIL screen

Sample screen

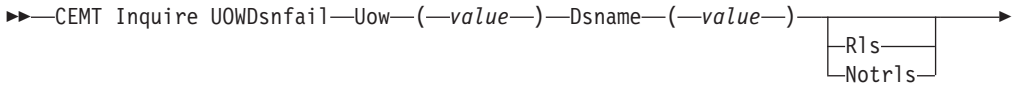
Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

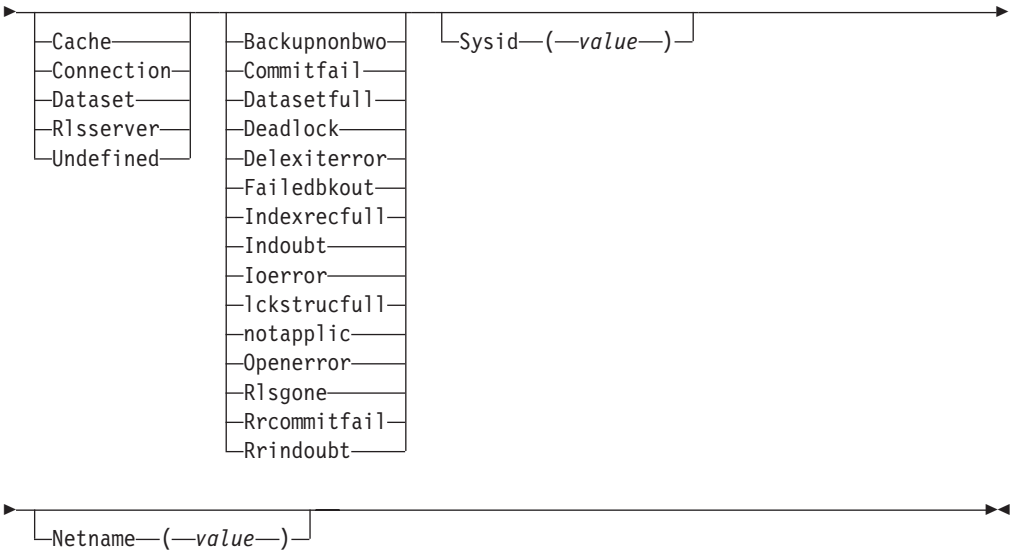
If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 163.

```
INQ UOWDSN
RESULT
  Uowdsnfail
  Dsname(RLS.ACCOUNTS.ESDS.DBASE1)
  Cause(Dataset)
  Reason(Delexiterror)
  Uow(AA6DB080C40CEE01)
  Sysid()
  Netname()
  Rlsaccess(Rls)
```

Figure 163. The expanded display of an individual entry

CEMT INQUIRE UOWDSNFAIL





Displayed fields

Cause
displays which failed component has caused the UOW to have retained locks for this data set. The values are:

Cache
A cache structure, or connection to it, has failed.

Connection
There is an intersystem connection error, which has caused the UOW to fail while in-doubt. The name of the system to which connectivity was lost is returned on the SYSID parameter and its netname is returned on the NETNAME parameter. CICS returns additional information in the REASON parameter about the CONNECTION failure.

Dataset
The backout of a UOW has failed for this data set. The reason for the data set failure is returned in the REASON parameter.

RIsserver
The SMSVSAM server has failed. The reason for the data set failure is returned in the REASON parameter.

Undefined
The UOW is probably in-flight following an emergency restart.

Dsname (*value*)
displays the 44-character data set name of a data set that has retained locks for this UOW.

Netname (*value*)
displays the 8-character netname (when the cause is CONNECTION) of the remote system to which connectivity has been lost.

Reason
displays, when the cause is RLSSERVER, CONNECTION, or DATASET, the specific reason for the error against this data set. The values are:

Backupnonbwo
A non-BWO backup was in progress while the UOW was being backed out. When the backup completes, CICS automatically retries the UOW.

Commitfail
An error occurred at some point when the locks were in the process of

being released. This is an error that can normally be resolved by recycling the server (which should happen automatically). The locks were acquired as a result of recoverable requests having been issued against the data set.

Datasetfull
No space is available on the direct access device for adding records to a data set. You need to reallocate the data set with more space. You can then retry the backout using SET DSNAME RETRY.

See Moving recoverable data sets that have retained locksin the *CICS Recovery and Restart Guide*.

Deadlock (*non-rls data sets only***)**
A deadlock was detected. This may go away if the UOW is retried.

Delexiterror
Backout of a write to an ESDS failed because a logical delete global user exit program was not enabled, or a logical delete global user exit program decided not to execute the logical delete.

Failedbkout
This occurs as a result of a severe error being identified, and is possibly an error in either CICS or VSAM. The problem may go away if the UOW is retried. Note that CICS performs some first-failure data capture (FFDC) at the point where the error is first detected.

Indexrecfull
A larger alternate index record size needs to be defined for the data set. See Moving recoverable data sets that have retained locksin the *CICS Recovery and Restart Guide*.

This error can also occur when a unique alternate index key, for a non-RLS data set, has been reused and CICS is now backing out the request which had removed that key value.

Indoubt
The unit of work had issued recoverable requests against the data set, and has now failed in-doubt. The connection to the coordinating system needs to be reestablished.

Ioerror
A hard I/O error occurred. To correct this error, restore a 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 RETRY option of the SET DSNAME 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 value 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 RLS server is down. 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 whilst 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 indoubt. 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 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 corresponds to only one Cause value, as shown in the following table:

Cause	Reason
CACHE	NOTAPPLIC
CONNECTION	INDOUBT
CONNECTION	RRINDOUBT
DATASET	BACKUPNONBWO
DATASET	DELEXITERROR
DATASET	DATASETFULL
DATASET	DEADLOCK
DATASET	FAILEDBKOUT
DATASET	INDEXRECFULL
DATASET	IOERROR
DATASET	LCKSTRUCFULL
DATASET	OPENERROR
RLSSERVER	COMMITFAIL
RLSSERVER	RLSGONE
RLSSERVER	RRCOMMITFAIL
UNDEFINED	Not applicable

Rlsaccess
displays whether the data set was last opened in this CICS region in RLS or non-RLS mode. The values are:

Rls The last open in this CICS region was in RLS mode.

Notrls
The last open in this CICS region was in non-RLS mode.

Sysid(*value***)**
displays the 4-character sysid (when the cause is CONNECTION) of the remote system to which connectivity has been lost.

Uow (*value***)**
displays the 16-character local identifier of the associated unit of work.

UOWDsnfail
indicates that this panel relates to an inquiry about the cause of a UOW failure.

CEMT INQUIRE UOWENQ

Retrieve information about enqueues held or waited on by a UOW, or about UOWs holding or waiting on a specified enqueue.

Description

INQUIRE UOWENQ returns information about the enqueues held by a UOW. 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.

You can use INQUIRE UOWENQ to indicate which records and queues would be affected if the UOW were forced.

The transaction also returns information about UOWs that are waiting on the enqueues. This enables you to diagnose enqueue deadlocks between tasks wanting to update the same resources.

Enqueues are typically held in active state, which means that other tasks are allowed to wait for the enqueue. However, if the UOW that owns the enqueue suffers an in-doubt failure, the enqueue is usually converted to the retained state until the in-doubt failure can be resolved. INQUIRE UOWENQ also retrieves information about these enqueues and can be used to identify which records and queues would be affected if the UOW were forced.

Note:

1. 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.)
2. 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.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE UOWENQ` (the minimum abbreviation is `CEMT I UOWE`). You get a display that lists all enqueues for all the current UOWs.
- Type `CEMT INQUIRE UOWENQ` (`CEMT I UOWE`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i uowe d`, the resulting display will show you the details only of enqueues on records in VSAM data sets open in non-RLS mode or CICS-maintained data tables.

Sample screen

```
INQ UOWE
STATUS:  RESULTS
 Uow(AC143001D4A72601) Tra(CECI) Tas(0000038) Act Exe Own
    Res(ACCOUNTS.DBASE1           ) Rle(015) Enq(00000000)
 Uow(AC143001D4A72601) Tra(CECI) Tas(0000038) Act Add Own
    Res(g...                      ) Rle(004) Enq(00000000)
```

Figure 164. CEMT INQUIRE UOWENQ screen

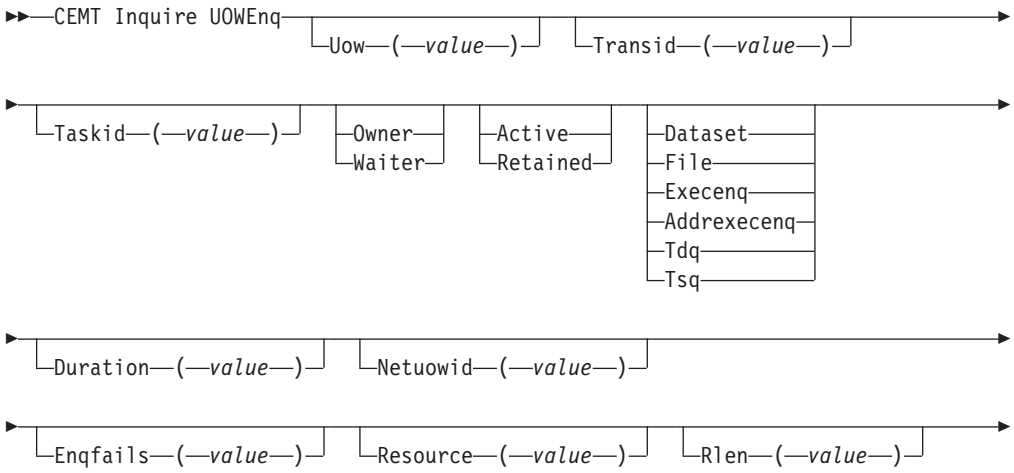
Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

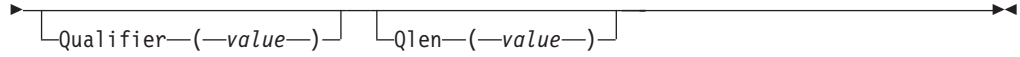
If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 165.

```
INQ UOWE
RESULT
  Uowenq
  Uow(AC143001D4A72601)
  Transid(CECI)
  Taskid(0000038)
  State(Active)
  Type(Execenq)
  Relation(Owner)
  Resource(ACCOUNTS.DBASE1)
  Rlen(015)
  Enqfails(00000000)
  Netuowid(..GBIBMIYA.IYA2T5C2...Mx...)
  Qualifier()
  Qlen(000)
```

Figure 165. The expanded display of an individual entry

CEMT INQUIRE UOWENQ





Displayed fields

Duration(*value***)**
displays the elapsed time (in seconds) since the enqueue entered its current state of OWNER, WAITER or RETAINED.

Enqfails (*value***)**
displays, for enqueues held in retained state, the number of failed enqueue attempts for this resource since the enqueue was last acquired. This indicates how many UOWs have received LOCKED responses due to this enqueue. The ENQFAILS option helps you identify which UOWs are causing “bottlenecks”.

For enqueues held in action state, ENQFAILS displays zero.

Enqscope (*value***)**
If the enqueue has sysplex scope, ENQSCOPE displays 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 displays blanks.

All CICS systems with the same ENQSCOPE value share the same sysplex Enqueue name space.

Netuowid(*value***)**
displays a 27-character string containing the network-wide LU6.2 name of the UOW that owns or is waiting for the enqueue.

Note: Nondisplayable characters appear as periods. You can use PF2 on the expanded panel: to see the value in hexadecimal.

Qlen (*value***)**
displays the length of the data, in the range 0 through 255, displayed in the QUALIFIER field.

Qualifier (*value***)**
displays a 1- through 255-character qualifier (for example, record id) that further identifies the resource associated with the enqueue lock. The data displayed depends on the type of enqueue, as shown in Table 4 on page 336. If no QUALIFIER data applies (that is for EXECENQ, ADDREXECENQ, or TSQ), a value of zero is displayed.

Relation
displays whether the data is associated with the owner of the enqueue or with a task waiting for the enqueue. The values are:

Owner
The (*value*), NETUOWID, TASKID, and TRANSID are those of the owner of the enqueue.

Waiter
The (*value*), NETUOWID, TASKID, and TRANSID are those of a waiter for the enqueue.

Resource (*value***)**
displays a 1- through 255-character string containing the name of the resource associated with the enqueue lock. The data displayed depends on the type of enqueue, as shown in Table 4 on page 336.

Note: Nondisplayable characters appear as periods. You can use PF2 on the expanded panel: to see the value in hexadecimal.

Rlen (*value*)
displays the length of the data, in the range 0 through 255, displayed in the RESOURCE field.

State
displays the state that the enqueue is held in. The 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 (*value*)
displays the number of the task associated with the UOW. If the UOW is shunted, it is the task number associated with the UOW before it was shunted.

Transid (*value*)
displays the 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
displays the type of resource being enqueued upon. The values are:

Dataset
The resource is a record in a VSAM data set (or a CICS-maintained data table). RESOURCE contains the name of the data set and QUALIFIER contains the record identifier.

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.

Execenq
The resource is associated with an EXEC CICS ENQ request. RESOURCE contains the enqueue argument passed on the request.

Addrexecenq
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)

Tdq 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 a read or write lock is held for the queue.

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.

Tsq The resource is a recoverable temporary storage queue. RESOURCE contains the name of the queue.

The data displayed in the RESOURCE and QUALIFIER fields depends on the resource type, as shown in Table 4.

Table 4. Data displayed in RESOURCE and QUALIFIER

Type	Resource	Qualifier
DATASET	Dataset name	Record identifier

Table 4. Data displayed in RESOURCE and QUALIFIER (continued)

Type	Resource	Qualifier
EXECENQ	EXEC enqueue argument	None
ADDREXECENQ	Address of EXEC enqueue argument	None
FILE	File name	Record identifier
TDQ	TD queue name	FROMQ or TOQ
TSQ	TS queue name	None

Uow(*value***)**
displays the 16-character local identifier of the associated unit of work.

CEMT INQUIRE UOWLINK

Retrieve information about connections involved in units of work.

Description

INQUIRE UOWLINK returns information about a connection involved in a unit of work. The connection can be to a remote system, or to a task-related user exit. If it is to a remote system, INQUIRE UOWLINK displays 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 displays the exit's entry name and qualifier.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE UOWLINK` (the minimum abbreviation is `CEMT I UOWL`). You get a display that lists all the current UOW-links.
- Type `CEMT INQUIRE UOWLINK` (`CEMT I UOWL`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i uowl rmi`, the resulting display will show you details only of UOW-links to external resource managers.

(*value***)**
An 8-byte token identifying the UOW-link for which data is to be returned.

ALl
All is the default.

Sample screen

```
I UOWL
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Uowl(01840009) Uow(AC142A977E6F3204) Con Lin(IYLX1   )
    Coo Appc War Sys(ISC1)               Net(..GBIBMIYA.IYCWTC13..p...
 Uowl(01C40006) Uow(AC142A977E6F3204) Con Lin(IYLX3   )
    Sub Irc  War Sys(MRO3)               Net(..GBIBMIYA.IYCWTC13..p...
```

Figure 166. CEMT INQUIRE UOWLINK screen

Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You might find it more

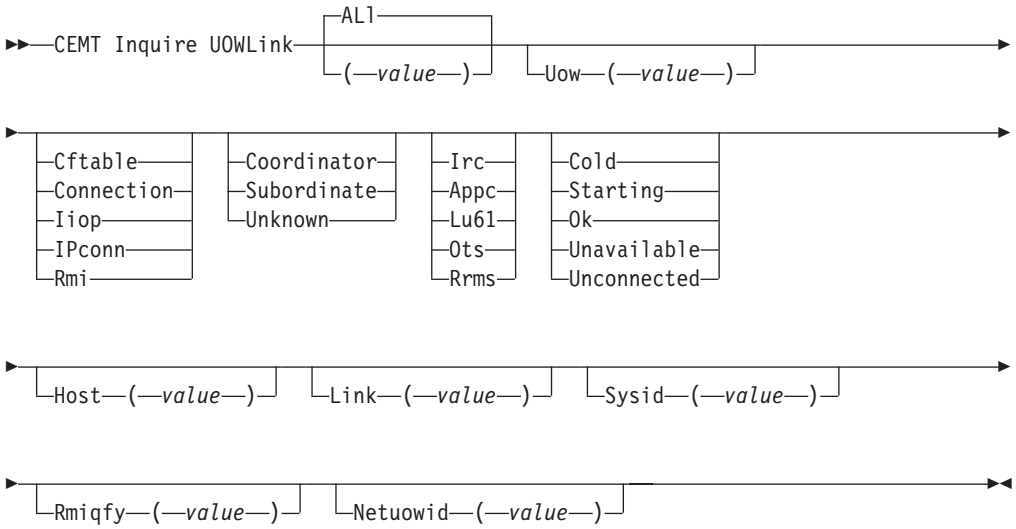
convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 167.

```
I UOWL
RESULT - OVERTYPE TO MODIFY
  Uowlink(01840009)
  Uow(AC142A977E6F3204)
  Type(Connection)
  Host( )
  Link(IYLX1)
  Action(              )
  Role(Coordinator)
  Protocol( )
  Resyncstatus( )
  Sysid( )
  Rmiqfy( )
  Netuowid( )
  Urid
  Xid
```

Figure 167. The expanded display of an individual entry

CEMT INQUIRE UOWLINK



Displayed fields

Host(*value***)**
For TYPE values IIOP and IPCONN, displays a 255-character area into which is returned, the TCP/IP host name, or a string containing the dotted decimal TCP/IP address, used to refer to the participant in the OTS transaction. Strings with fewer than 255 characters are padded with blanks. This is useful for identifying the participant especially when problems occur. Note that the UOW and the participant might belong to the same CorbaServer.

For other TYPE values, HOST displays blanks.

Link (*value*)
The value returned depends on the TYPE returned:

CFTABLE
The 8-character name of the coupling facility data table pool

CONNECTION
The 8-character netname of the remote system.

IIOP
Blanks. The equivalent IIOP information is returned by the HOST attribute.

IPCONN
The 8-character APPLID of the remote system.

RMI
The entry name of the task-related user exit.

Netuowid (*value*)
Displays a 27-character string containing the network-wide LU6.2 name for the UOW.

Note: Nondisplayable characters appear as periods. You can use PF2 on the expanded panel to see the value in hexadecimal.

Protocol
Displays the communication protocol used by the connection. The values are:
Irc Inter-Region Communication. This is an MRO connection.
Appc Advanced Program to Program Communication.
Lu61 LUTYPE 6.1.
IPic IPIC connection.
Ots Object Transaction Services for Java programs.
Rrms Indicates that the UOW is coordinated by RRS/MVS.

Resyncstatus
Displays the resynchronization status of the connection. The values are:
Cold The connection has been cold started by the partner system. The partner can no longer coordinate any in-doubts for this system; nor can this system pass to the partner any decisions remembered for it.
Starting
The connection is being acquired, but the exchange lognames process has not yet completed.
Ok The connection is operating normally. If there has been a failure of the partner system, the partner has been restarted and the connection able to resynchronize the associated UOW.
Unavailable
The connection is not currently acquired.
Unconnected
There is no associated connection.

Rmiqfy (*value*)
Displays, for a connection type of RMI, the 8-character entry qualifier of the task-related user exit. For all other connection types, this value displays blank.

Role
Displays the role of the connection. The 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 (*value*)
Displays, for a TYPE value of CONNECTION, the 4-character SYSID of the connection. If the connection has been discarded, or the connection-type is RMI, IPCONN, CFTABLE or IIOP, or the PROTOCOL field is showing RRMS, SYSID displays blanks.

Type
Displays the type of connection. The values are:
Cftable
A connection to a CFDT server.
Connection
A connection defined by a CONNECTION resource definition.
Iiop An OTS link.
IPconn
A connection defined by an IPCONN resource definition.
Rmi A connection to an external resource manager using the resource manager interface (RMI).

UOWLink(*value*)
Indicates that this panel relates to an inquiry about a UOW-link and displays the 8-character UOW-link identifier.

Uow (*value*)
Displays the 16-character local identifier of the associated unit of work.

Urid (*value*)
If the PROTOCOL field displays RRMS, this field contains the 32-byte hexadecimal representation of the RRMS unit of recovery identifier. For other values of the PROTOCOL field (including OTS), URID displays blanks.

Xid(*data-area*)
Displays the 64-character global transaction identifier, if present.

CEMT INQUIRE URIMAP

Retrieve information about URIMAP resource definitions.

Description

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.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE URIMAP` (the minimum abbreviation is `CEMT I UR`). You get a display that lists the current status.
- Type `CEMT INQUIRE URIMAP` (`CEMT I UR`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i ur tcpipservice`, the resulting display will show you the details of only those URIMAP resource definitions that apply to the selected TCPIPSERVICE definition (which defines a port).

To change various attributes, overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 128).

ALl
is the default.

(*value***)**
is the name of the URIMAP definition for which information is requested.

Sample screen

```
I URIMAP
STATUS:  RESULTS
 Uri(B13PASS ) Cli Ena     Http
     Host(www.hursleytest.ibm.com:8080) Path(/INDEX.HTML                  )
  Uri(B24PASS ) Ser Ena     Http
     Host(www.hursleytest.ibm.com     ) Path(/                            )
  Uri(B26PASS ) Ser Ena     Http
     Host(*                           ) Path(/                            )
```

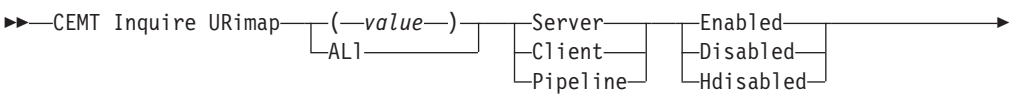
Figure 168. CEMT INQUIRE URIMAP screen

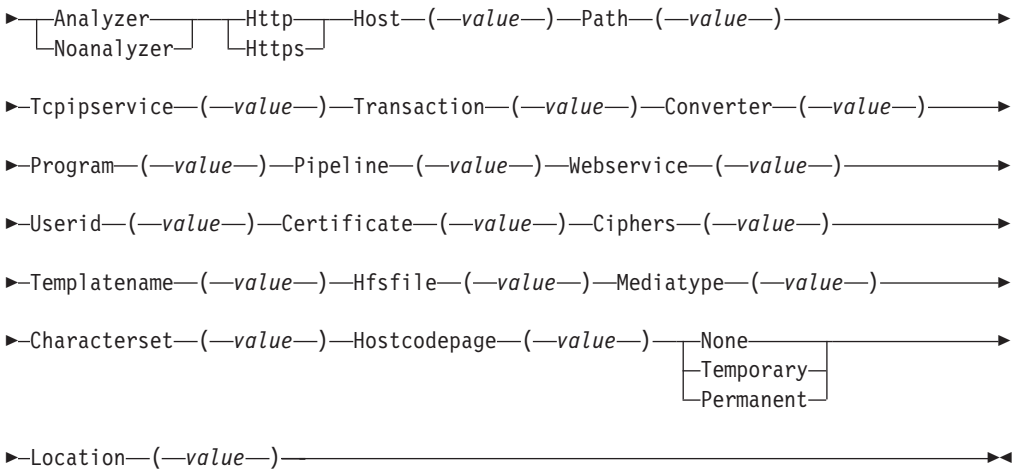
If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 169.

```
 I URIMAP
 RESULT
  Urimap(B13PASS)
   Usage(Client)
   Enablestatus( Enabled )
   Analyzerstat(Noanalyzer)
   Scheme(Http)
   Host(www.hursleytest.ibm.com:8080)
   Path(/INDEX.HTML)
   Tcpipservice()
   Transaction()
   Converter()
   Program()
   Pipeline()
   Webservice()
   Userid()
   Certificate()
   Ciphers()
   Templatename()
   Hfsfile()
   Mediatype()
   Characterset()
   Hostcodepage()
   Redirecttype(None)
   Location(                                                          )
   Location(                                                          )
   Location(                                                          )
   Location(                                                          )
   Location(         )
```

Figure 169. The expanded display of an individual entry

CEMT INQUIRE URIMAP





Displayed fields

Usage
displays the usage for the URIMAP definition. The possible values are:

Server
The URIMAP definition is for CICS as an HTTP server. It is used to map the URL of an incoming HTTP request from a Web client, to CICS application resources.

Client
The URIMAP definition is for CICS as an HTTP client. It is used when CICS makes a request for an HTTP resource on a remote server, so that you can avoid identifying the URL in your application program.

Pipeline
The URIMAP definition is for a Web service. It is used to specify the processing that is to be performed on a request by which a client invokes a Web service in CICS.

Enablestatus
displays the status of the URIMAP definition. The possible values are:

Enabled
The URIMAP definition can be accessed.

Disabled
The URIMAP definition cannot be accessed. A URIMAP definition with this status can be deleted.

Hdisabled
The URIMAP definition cannot be accessed because the virtual host of which it forms a part has been disabled. Use CEMT SET HOST to re-enable all the URIMAP definitions in the virtual host. A URIMAP definition with this status cannot be deleted.

Analyzerstat(*value***)**
displays the analyzer program setting for the URIMAP definition. The possible values are:

Analyzer
The analyzer associated with the TCPIPSERVICE definition is to be used in processing the HTTP request.

Noanalyzer
The analyzer program is not to be used.

Scheme
displays the scheme component of the URL. The possible values are:

HTTP
HTTP without Secure Sockets Layer (SSL).

HTTPS
HTTP with Secure Sockets Layer (SSL).

Host(*value***)**
displays the host component of the URL, which can be up to 116 characters.

Path(*value***)**
displays the path component of the URL, which can be up to 256 characters.

Tcpipservice(*value***)**
displays the 8-character name of the TCPIPSERVICE definition that specifies the inbound port to which this URIMAP definition relates. If this definition is not given, the URIMAP definition relates to all TCPIPSERVICE definitions.

Transaction(*value***)**
displays the 4-character name of an alias transaction to run the user application that composes a response.

Converter(*value***)**
displays the 8-character name of a converter program that performs conversion or other processing on the content of the request.

Program(*value***)**
displays the 8-character name of the application program that composes a response.

Pipeline(*value***)**
displays the 8-character name of the PIPELINE resource definition that provides information about the processing nodes which will act on the service request from the client.

Webservice(*value***)**
displays 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. It defines aspects of the run time environment for a CICS application program deployed in a Web services setting.

Userid(*value***)**
displays the 8-character user ID under which the alias transaction will be attached.

Certificate(*value***)**
displays 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, which can be up to 32 characters.

Ciphers(*value***)**
displays the cipher code that is to be used for the HTTP request by CICS as an HTTP client, which can be up to 48 characters.

Templatename(*value***)**
displays the name of a CICS document template that is used to form the static response, which can be up to 40 characters.

Hfsfile(*value*)
displays the fully qualified name of a UNIX System Services file that will form the static response, which can be up to 256 characters.

Mediatype(*value*)
displays a description of the data content of the static response, which can be up to 40 characters.

Characterset(*value*)
displays the name of the character set for the static response, which can be up to 40 characters.

Hostcodepage(*value*)
displays the IBM codepage (EBCDIC) in which the text document that will form the static response is encoded; this can be up to 10 characters.

Redirecttype
displays the type of redirection for requests that match this URIMAP definition. The URL for redirection is specified by the Location option. The possible 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).

Location(*value*)
displays a URL to which matching HTTP requests from Web clients will be redirected, which can be up to 255 characters. Redirection is activated by the setting specified by the REDIRECTTYPE option.

CEMT INQUIRE VTAM

Inquire on the state of the connection between CICS and VTAM.

Description

INQUIRE VTAM inquires on the state of the connection between CICS and VTAM, and also on the generic resource name (GRNAME) and the persistent session delay interval (PSDI).

Input

Press the Clear key and type `CEMT INQUIRE VTAM` (the minimum abbreviation is `CEMT I V`). You will get a display screen, a sample of which is shown below.

To change various attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 128).
- Use the CEMT SET VTAM command.

Sample screen

```
IN VT
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Vta Ope Psd( 000000 )
```

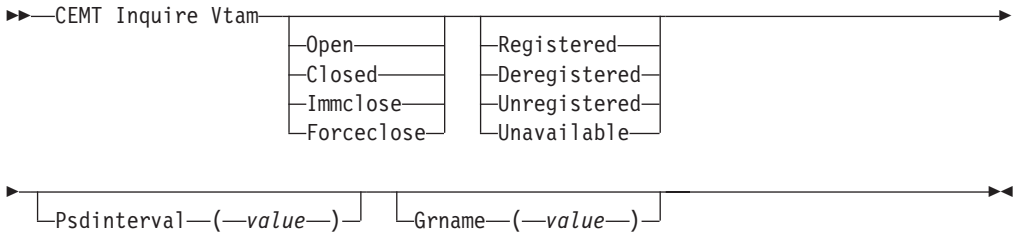
Figure 170. CEMT INQUIRE VTAM screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 171.

```
IN VT
RESULT - OVERTYPE TO MODIFY
  Vtam
  Openstatus( Open )
  Psdinterval( 000000 )
  Grstatus()
  Grname()
```

Figure 171. The expanded display

CEMT INQUIRE VTAM



Displayed fields

Grname(*value***)**
displays the 8-character generic resource name under which this CICS region is registered to VTAM.

Grstatus
displays the status of generic resource registration. The values are:

Registered
CICS registered successfully as a member of the VTAM generic resource named by the GRNAME option.

Deregistered
CICS was registered as a member of the generic resource named by the GRNAME and has deregistered successfully.

Unregistered
A generic resource name (GRNAME) was specified, but CICS has not made any attempt to register.

Unavailable
A generic resource name (GRNAME) was specified, but the version of VTAM that CICS is connected to does not support generic resources.

If CICS makes an unsuccessful attempt to register, the status is shown as UNREGISTERED, and the words REGISTER ERROR appear alongside.

If CICS makes as unsuccessful attempt to deregister, the status is shown as REGISTERED, and the words DEREGISTER ERROR appear alongside.

Openstatus
displays the status of the connection between CICS and VTAM. The values are:
Open There is a connection between CICS and VTAM.
Closed
The connection between CICS and VTAM has terminated.

Note: You can reset this value by overtyping it with a different value, or with:
Immclose
The connection between CICS and VTAM is in the process of closing following an EXEC CICS SET VTAM IMMCLOSE or CEMT SET VTAM IMMCLOSE command.
Forceclose
The connection between CICS and VTAM is in the process of closing following an EXEC CICS SET VTAM FORCECLOSE or CEMT SET VTAM FORCECLOSE command.

Psdinterval(*value***)**
displays the persistent session delay interval, expressed in the form *hhmmss*. It specifies if and for how long sessions are held in recovery-pending state after CICS fails.

Note: You can reset this value by overtyping it with a different value.
The permitted range is 000000 to 235959 inclusive.
000000
If CICS fails, sessions are terminated.
Hhmmss
If CICS fails, sessions are held in recovery-pending state by VTAM for up to the interval specified.

Vtam
indicates that this panel relates to a VTAM inquiry.

CEMT INQUIRE WEB

Retrieve information about CICS Web support.

Description

INQUIRE WEB returns information about the state of CICS Web support.

Input

Press the Clear key and type `CEMT INQUIRE WEB` (the minimum abbreviation is `CEMT I WE`). You will get a display screen, a sample of which is shown below.

To change various attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 128).
- Use the CEMT SET WEB command.

Sample screen

```
IN WE
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Web Gar( 00060 ) Tim( 00005 ) Loc
```

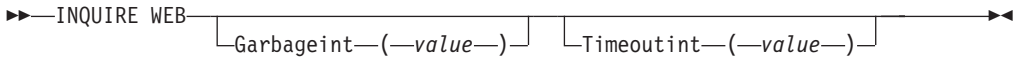
Figure 172. CEMT INQUIRE WEB screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 171 on page 345.

```
I WEB
RESULT - OVERTYPE TO MODIFY
  Web
  Garbageint( 00060 )
  Timeoutint( 00005 )
```

Figure 173. The expanded display

CEMT INQUIRE WEB



Displayed fields

Garbageint (*cvda***)**
is the interval at which the Web garbage collection task runs to clean up Web 3270 state data for which the terminal timeout interval has expired. The interval is measured in minutes.

Timeoutint (*cvda***)**
is the period of time after which inactive Web 3270 sessions are eligible for garbage collection. The period is measured in minutes.

CEMT INQUIRE WEBSERVICE

Retrieve information about an installed WEBSERVICE.

Description

Use the **INQUIRE WEBSERVICE** command to retrieve information about an installed WEBSERVICE.

Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE WEBSERVICE` (the minimum abbreviation is `CEMT I WEBS`). You get a screen that lists the current status.
- Type `CEMT INQUIRE WEBSERVICE` (`CEMT I WEBS`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i Webs en`, the resulting display will show you the details of only those process-types that are enabled.

To change various attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field.
- Use the **CEMT SET WEBSERVICE** command.

ALl
is the default. Information about all WEBSERVICEs is displayed, unless you specify a selection of WEBSERVICEs to be queried.

(value)
is the name (1-8 characters) of an installed WEBSERVICE definition.

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format of all the attributes for the specified WEBSERVICE.

Displayed fields

Binding*(value)*
Displays the WSDL binding represented by the WEBSERVICE. This binding is one of (potentially) many that appear in the WEBSERVICE configuration file associated with the WEBSERVICE resource. The name can be up to 255 characters long.

CCSID(*value*)
Displays the CCSID that is used to encode data between the application program and the Web service binding file at run time. This value is set using the optional **CCSID** parameter in the Web services assistant when the Web service binding file was generated. If the *value* is 0, the default CCSID for the CICS region that is specified by the **LOCALCCSID** system initialization parameter is used.

Container*(value)*
Displays the name of the container used if PGMINTERFACE returns a value of CHANNEL. The name can be up to 16 characters long.

Endpoint*(value)*
Displays 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(*value*)
Displays the time that the deployed WSBind file on z/OS UNIX was last 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 WEBSERVICESs (those installed by the CICS scanning mechanism), the value of LASTMODTIME is the timestamp of the z/OS UNIX file pointed to by the WSBind definition, at the time the WEBSERVICE definition was last installed or updated. For statically-installed WEBSERVICESs (those installed from a CSD or by CREATE WEBSERVICE), the value of LASTMODTIME is the timestamp of the WSBind z/OS UNIX file pointed to by the WEBSERVICE definition, at the time 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.

Mappinglevel(*value*)
Displays the mapping level that is used to convert data between language

structures and Web service description (WSDL) documents. The value of the mapping level is 1.0, 1.1, 1.2, 2.0 or 2.1. The default is to use a mapping level of 1.0.

Minrunlevel(*value*)
Displays the minimum runtime level that is required to run the Web service in CICS. The value of the runtime level is 1.0, 1.1, 1.2, 2.0 or 2.1.

Pgminterface
Displays an indication of 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*(value)*
Displays 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*(value)*
Displays 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.

Urimap*(value)*
Displays 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
Indicates whether full validation of SOAP messages is currently enabled for this WEBSERVICE:

Validation
Full validation is enabled

Disabled
Full validation is disabled

WSBind*(value)*
Displays the name of the Web service binding file. The name can be up to 255 characters long.

WSDlfile*(value)*
Displays the name of the Web service description file associated with the WEBSERVICE resource. The name can be up to 255 characters long.

Xopdirectst*(value)*
Indicates whether the Web service is capable of handling XOP documents and binary attachments in direct mode.

NOXOPDIRECT
The Web service cannot handle XOP documents and binary attachments in direct mode. This is either because validation is switched on for the Web

service, or because the Web service implementation does not support the handling of XOP documents and binary attachments. Compatibility mode is used instead.

XOPDIRECT
The Web service can handle XOP documents and binary attachments in direct mode.

Xopsupportst*(value)*
Indicates whether the Web service implementation is capable of handling XOP documents and binary attachments.

NOXOPSUPPORT
The Web service implementation is not capable of handling XOP documents and binary attachments.

XOPSUPPORT
The Web service implementation is capable of handling XOP documents and binary attachments. This is true for any CICS-generated web service created by a level of CICS that supports MTOM/XOP.

CEMT INQUIRE WORKREQUEST

Retrieve information about work requests in the local CICS region.

Description

INQUIRE WORKREQUEST returns information about the various EJB tasks that are started as a result of action by a request receiver. This allows you to track these tasks. In this topic we often refer to tasks of this sort as work requests.

Tracking EJB tasks is superficially not unlike tracking the various tasks of a distributed UOW, however, because it applies to the Enterprise bean and IIOP environment, you need different tools to manage it.

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/RequestProcessor relationship is not the same as any other type of transaction relationship e.g. the web. 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. You can

- determine which transactions are associated with this work request
- correlate all transactions associated with this work request (for example, for accounting purposes)

The INQUIRE WORKREQUEST command is based on the type of work being carried out. The only type supported in this release is IIOP.

Work Requests are identified by tokens. Each token is unique in the local system for the lifetime of the request, but subsequently, it can be reused.

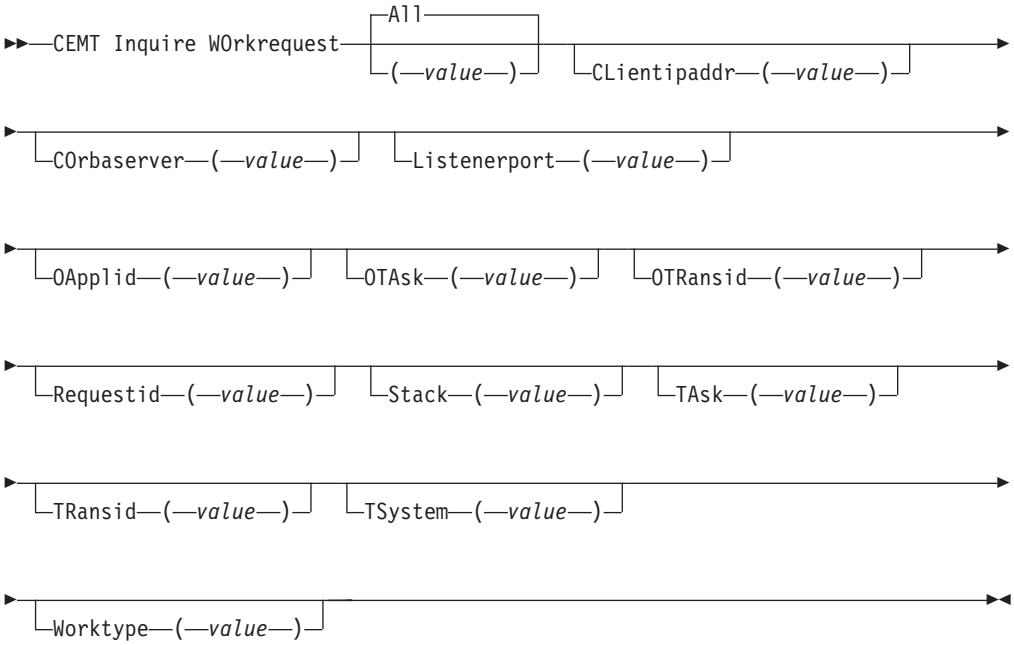
Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT INQUIRE WORKREQUEST` (the minimum abbreviation is `CEMT I WO`). You get a display that lists the current status.
- Type `CEMT INQUIRE WORKREQUEST` (`CEMT I WO`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i wo co(MJW1)`, the resulting display will show you the details of only those work requests associated with the CORBASERVER definition named MJW1.

You cannot change the attributes displayed. Use the CEMT SET WORKREQUEST command to specify PURGE or FORCEPURGE for a particular work request.

CEMT INQUIRE WORKREQUEST



(*value***)**
An 8-character token to identify the work request. The valid characters for this token are the 16 hexadecimal digits, 0–9 and 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.

All
is the default. As many work requests are displayed as are present in the system.

You cannot specify a list of identifiers, nor can you use the symbols * and + to specify a family of work requests.

Sample screen

```
INQUIRE WORKREQUEST
STATUS:  RESULTS - OVERTYPE TO MODIFY
Wor(01A54E3B) Iio Tas(0000090) Tra(CIRP)
   Cli(9.20.7.123     ) Lis(20142) Cor(CHN1) Oap(IYK2ZFV1) Ota(0000101)
Wor(15BED280) Iio Tas(0000090) Tra(CIRP)
   Cli(9.20.7.123     ) Lis(20142) Cor(CHN1) Oap(IYK2ZFV1) Ota(0000101)
Wor(15BED6E0) Iio Tas(0000176) Tra(CIRP)
   Cli(9.20.7.123     ) Lis(20142) Cor(CHN1) Oap(IYK2ZFV1) Ota(0000251)
Wor(15BEDBC0) Iio Tas(0000176) Tra(CIRP)
   Cli(9.20.7.123     ) Lis(20142) Cor(CHN1) Oap(IYK2ZFV1) Ota(0000251)


                                                SYSID=JOHN APPLID=IYK2ZFV1
   RESPONSE:  NORMAL                          TIME:  11.26.40  DATE: 08.11.02
 PF 1 HELP        3 END       5 VAR       7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 174. CEMT INQUIRE WORKREQUEST screen

When the character P or F (Purge or Forcepurge) is entered in the action column CEMT initiates a SET WORKREQUEST command.

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 175.

```
   INQUIRE WORKREQUEST (Expansion of one line from above)
   RESULT - OVERTYPE TO MODIFY
     Workrequest(01A54E3B)
     Worktype(Iiop)
     Task(0000013)
     Transid(CIRP)
     Purgetype(              )
     Clientipaddr(9.20.7.123)
     Listenerport(20142)
     Corbaserver(GMB1)
     Oapplid(IYK2ZFV1)
     Otask(0000101)
     Otransid(CIRR)
     Tsystem(9.20.101.720)
     Requestid(0000010032)
     Stack(0000002)
                                                SYSID=JOHN APPLID=IYK2ZFV1
                                              TIME:  11.26.40  DATE: 08.11.02
 PF 1 HELP 2 HEX 3 END       5 VAR        7 SBH 8 SFH       10 SB 11 SF
```

Figure 175. The expanded display of an individual entry

Displayed fields

CLientipaddr(*value***)**
displays the TCPIP address of the client that originated the request as a 15-character value.

COrbaserver(*value***)**
displays the Name of the CorbaServer as a 4-character value.

Listenerport(*value***)**
displays the Host port that received the request, as a 5-digit numeric value.

OApplid(*value***)**
displays the applid of the originating task as an 8-character value.

OTAsk(*value***)**
displays the number of the task (RequestReceiver) that received the request, as a 7-digit numeric value.

OTRansid(*value*)
displays as a 4-character value, the transaction id that received the request.

Purgetype (*input only field*)
specifies whether a task is to be purged or forcepurged. The values are:

Forcepurge
The work request is terminated immediately. System integrity is not guaranteed. In some extreme cases, for example if a work request is forcepurged during backout processing, CICS terminates abnormally. If you want to terminate a work request but do not want to terminate CICS, you should use PURGE instead of FORCEPURGE.

Purge
The work request is terminated, but termination occurs only when system and data integrity can be maintained.

Requestid(*value*)
displays the current request as a 7-digit numeric value. This value is sent by the client initiating the request and is used to associate the reply with the request. A client can be processing more than one request at any given moment.

Stack(*value*)
displays the stack level of this call as a 7-digit numeric value.

As each new request is received from the client the request receiver creates a stack and initializes it with a value of '1'. The stack is incremented by '1' every time a new bean is invoked, and decremented each time a bean which has incremented it terminates. So the value of the stack gives the current level within the EJB transaction at which this work request is executing. If a request is sent to a non-CICS EJB server then this field is NOT incremented.

TAsk(*value*)
displays the Local task number, as a 7-character numeric value. It is translated to packed decimal, or in C to an unsigned char(4).

TRansid(*value*)
displays the Local transaction id - 4 character

TSystem(*value*)
displays the applid of the target CICS system as a 21-byte field. It 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
- 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

TSYSTEM is needed when going outbound over TCPIP. It is the only identification of the target system that is available.

WOrkrequest(*value*)
displays an 8-character token to identify the work request. The valid characters for this token are the 16 hexadecimal digits, 0–9 and 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(*value*)
displays the type of work being performed.

IIOP Specifies that the work is being performed for an IIOP request.

SOAP Specifies that the work is being performed for a Web service request.

CEMT PERFORM commands

The CEMT PERFORM commands are not concerned with resource status. They are used to:

- Initialize and terminate the shared class cache, and set its attributes (PERFORM CLASSCACHE)
- Publish or retract all the beans installed in a CorbaServer; or scan a CorbaServer's deployed JAR file directory for new or updated deployed JAR files (PERFORM CORBASERVER)
- Cause the CICS timeout delete mechanism to be invoked immediately (PERFORM DELETSHIPPED)
- Publish or retract beans installed in a deployed JAR file (PERFORM DJAR)
- Dump and continue processing (PERFORM DUMP or PERFORM SNAP)
- End an affinity owned by CICS (PERFORM ENDAFFINITY)
- Scan a PIPELINE's Web service binding directory for updated Web service binding files (PERFORM PIPELINE)
- Reset the date and time of day (PERFORM RESET)
- Rebuild the RACF resource profile (PERFORM SECURITY REBUILD)
- Shut down the CICS region (PERFORM SHUTDOWN)
- Request statistics to be recorded (PERFORM STATISTICS RECORD).

Descriptions of the CEMT PERFORM commands follow in alphabetic order. The operands for each of the commands are also presented in alphabetic order.

CEMT PERFORM CLASSCACHE

You can use the PERFORM CLASSCACHE command to initialize (start or reload) and terminate (phase out, purge or forcepurge) the shared class cache.

Description

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 if you are using the IBM SDK for z/OS, V1.4.2 to provide Java support, 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.

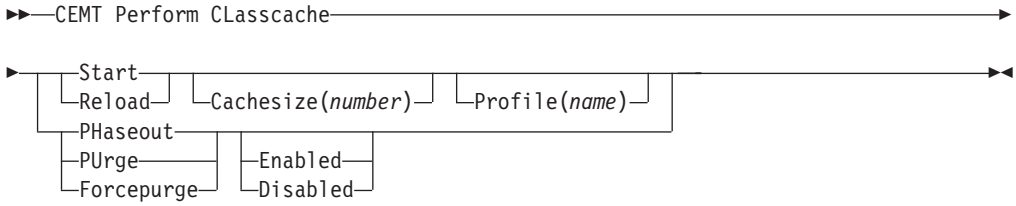
When you change the size, JVM profile or autostart status of the shared class cache while CICS is running, subsequent warm or emergency CICS restarts use the changed setting, unless the relevant system initialization parameter is specified

as an override at startup, in which case the behavior from the system initialization parameter is used. This does not affect the shared class cache provided by the IBM SDK for z/OS, V5 because it persists across warm or emergency restarts, but it does affect the Version 1.4.2 shared class cache. If the system is initial or cold started, the settings are always taken from the relevant system initialization parameters.

When you terminate (phase out, purge or forcepurge) the shared class cache, 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 CEMT PERFORM CLASSCACHE START command (or the equivalent EXEC CICS 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.

Syntax

CEMT PERFORM CLASSCACHE



Options

The operations you can perform with the PERFORM CLASSCACHE command are:

Forcepurge
stops and deletes the shared class cache. All tasks using JVMs that are using the shared class cache are terminated by the SET TASK FORCEPURGE mechanism, and the JVMs are terminated. The shared class cache is deleted when all the JVMs that were using it have been terminated.

Phaseout
stops and deletes the shared class cache. All 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 JVMs that were using it have been terminated.

When you have issued one command to terminate the shared class cache, you can repeat the CEMT PERFORM CLASSCACHE command to make a stronger attempt to purge the tasks using the JVMs (for example, by using Purge instead of Phaseout). 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.

Purge
stops and deletes the shared class cache. All tasks using JVMs that are using the shared class cache are terminated by the SET TASK PURGE mechanism, and the JVMs are terminated. The shared class cache is deleted when all the JVMs that were using it have been terminated.

Reload
reloads the shared class cache. Perform this operation to create a new class cache when the status of the shared class cache is STARTED (the option only works if the shared class cache has been started).

If specified, the CACHESIZE and for the Version 1.4.2 shared class cache, PROFILE options are used in the creation of the cache. JVMs, both those that are already allocated to tasks and those that are allocated to tasks after you issue the command, continue to use the existing shared class cache until the new shared class cache is ready. When the new shared class cache is ready, subsequent requests for JVMs are given a JVM that uses the new cache. These new JVMs are started as they are requested by applications, and they replace the JVMs that are using the old shared class cache. The JVMs that are using the old shared class cache are allowed to finish running their current Java programs, and then they are terminated. The old shared class cache is deleted when all the JVMs that are using it have been terminated.

Start
starts the shared class cache. Perform this operation to create a new class cache when the status of the shared class cache is STOPPED. If specified, the CACHESIZE and for the Version 1.4.2 shared class cache, PROFILE options are used in the creation of the cache. The shared class cache can only be started after at least one JVM has been created in the CICS region, so that CICS can identify the Java version that is required. CICS retains the Java version across a warm or emergency start. If you issue the PERFORM CLASSCACHE START command after a cold start and before any JVMs have been created, CICS validates the parameters but does not start the shared class cache.

The additional options that you can use on the PERFORM CLASSCACHE command to set attributes of the shared class cache are:

Cachesize(*number***)**
You can use CACHESIZE when you are performing the Start or Reload operations. CACHESIZE specifies the size, in bytes, of the shared class cache that is to be started, or the new shared class cache that is to be reloaded. You can specify the size in bytes, or in kilobytes (K), megabytes (M) or gigabytes (G). For example, specifying 20M sets the size of the shared class cache to 20 megabytes. The number you specify can be up to 20 digits. The minimum size that you can specify is 8M (8388608K).

Disabled
If you are performing the Phaseout, Purge or Forcepurge operations, you can use this option to set the status of autostart for the shared class cache to DISABLED.

When autostart is disabled, if the shared class cache has been stopped or has not yet been started on this CICS execution, an explicit PERFORM CLASSCACHE command is required to start it. If the status of the shared class cache is STOPPED and autostart is disabled, and CICS receives a request to run a Java application in a JVM whose profile requires the use of the shared class cache, the application cannot run.

Enabled
If you are performing the Phaseout, Purge or Forcepurge operations, you can use this option to set the status of autostart for the shared class cache to ENABLED.

When autostart is 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.

Profile(*name***)**
You can use PROFILE when you are performing the Start or Reload operations.

This option specifies the eight-character name of the JVM profile that will be used for the master JVM that starts the shared class cache or loads the new shared class cache. If you are using the IBM SDK for z/OS, V5 for Java support, this option is ignored, because the Version 5 shared class cache does not have a master JVM.

When you use the name of a JVM profile anywhere in CICS, you must enter it using the same combination of upper and lower case characters that is present in the z/OS UNIX file name. If you need to enter the name of a JVM profile in mixed case when you are using the CEMT transaction, ensure that the terminal you use is correctly configured, with upper case translation suppressed.

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 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.

CEMT PERFORM CORBASERVER

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
- Scan the CorbaServer's deployed JAR file directory for new or updated deployed JAR files.

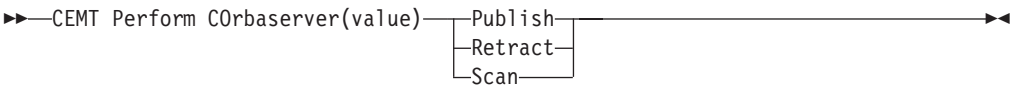
If no action is specified, the command has no effect. You can issue a **PERFORM CORBASERVER** command only when the CorbaServer is in a steady state—that is, when it is in ENABLED or DISABLED state, but *not* when it is in ENABLING, DISABLING, or DISCARDING state.

In addition to the traditional DFSMS-managed data sets, such as partitioned data sets (PDS), CICS requires access to z/OS UNIX System Services and its files in order to launch a JVM. This requires that the CICS region userid is authorized to z/OS UNIX System Services to enable CICS to access the required z/OS UNIX directories listed in the JVM profile and JVM properties file.

A failure to perform the requested action will be reported by returning a condition and RESP2 value from the command and, in certain cases, by writing messages to CSMT. The requested action may have been partially successful since a failure does not cause the effects of the command to be backed out. Also, failure does not necessarily stop the command from continuing in order to increase the extent to which it was partially successful.

Syntax

CEMT PERFORM CORBASERVER



Options

Corbaserver (*value*)
specifies the 4-character CorbaServer name.

Publish

publishes:

- all beans installed in the specified CorbaServer
- the Generic Factory Interoperable Object Reference (Generic factory) 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.

Retract
indicates that all the beans installed in the CorbaServer are to be retracted. 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.

Scan
scans the CorbaServer's deployed JAR file directory (also known as the *pickup directory*) for new or updated deployed JAR files.

If CICS finds any new deployed JAR files in the pickup directory, it copies them to its shelf directory and dynamically creates and installs DJAR definitions for them.

If CICS finds any updated deployed JAR files in the pickup directory, it updates both the LASTMODTIME, DATESTAMP, and TIMESTAMP attributes of the installed DJAR definitions and the shelf copies of the deployed JAR files.

Note: 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.

If you receive a "DJAR CONFLICT" message, it means that one or more deployed JAR files have failed to install due to a 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*.

CEMT PERFORM DELETSHIPPED

Cause the CICS timeout delete mechanism to be invoked immediately.

Description

The CICS timeout delete mechanism removes any shipped terminal definitions that have not been used for longer than the time specified on the DSHIPIDL system initialization parameter, or on a subsequent SET DELETSHIPPED IDLE command. The interval between scheduled invocations of the mechanism is specified on the DSHIPINT system initialization parameter, or on a subsequent SET DELETSHIPPED INTERVAL command.

PERFORM DELETSHIPPED invokes timeout delete immediately.

Note:

1. Shipped definitions are not deleted if there is an automatic initiate descriptor (AID) associated with the terminal.
2. Issuing a PERFORM DELETSHIPPED command does **not** reset the start of the time interval until the next invocation of the timeout delete mechanism. In other words, causing the timeout delete mechanism to be invoked immediately does not affect the time remaining until the next scheduled invocation.

Syntax

CEMT PERFORM DELETSHIPPED

CEMT PERFORM DJAR

The PERFORM DJAR command performs certain actions against a specified deployed JAR file. If no action is specified, the command has no effect.

Description

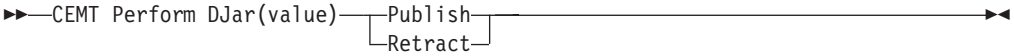
In addition to the traditional DFSMS-managed data sets, such as partitioned data sets (PDS), CICS requires access to z/OS UNIX System Services and its files in order to launch a JVM. This requires that the CICS region userid is authorized to

z/OS UNIX System Services to enable CICS to access the required z/OS UNIX directories listed in the JVM profile and JVM properties file.

A failure to perform the requested action will be reported by returning a condition and RESP2 value from the command and, in certain cases, by writing messages to CSMT. The requested action may have been partially successful since a failure does not cause the effects of the command to be backed out. Also, failure does not necessarily stop the command from continuing in order to increase the extent to which it was partially successful.

Syntax

CEMT PERFORM DJAR



Options

DJar (value)
specifies the 1–32 character name of a deployed JAR file.

Publish
indicates that all beans installed in the named deployed JAR file are to be published. 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 DJAR's CorbaServer and the name of the bean.

Retract
indicates that all the beans installed in the deployed JAR file are to be retracted. 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 DJAR's CorbaServer and the name of the bean.

CEMT PERFORM DUMP|SNAP

Obtain a CICS system dump.

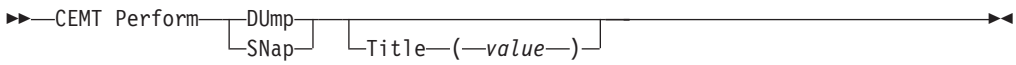
Description

To get a CICS system to dump and continue processing, you can use either the CEMT PERFORM DUMP command or the CEMT PERFORM SNAP command. Both commands produce a “snapshot” of the CICS system.

The use of CEMT PERFORM DUMP or CEMT PERFORM SNAP prevents all other tasks from running. Consequently, terminal operators may experience delays in response.

Syntax

CEMT PERFORM DUMP



Options

DUmp
specifies that a system dump is to be produced.

SNap
specifies that CICS is to issue the MVS SDUMP macro to produce an SDUMP (system dump) of the CICS region.

Title (*value*)
specifies a title of up to 32 characters; it is added at the beginning of the dump. If your title includes spaces, you must enclose the whole title within single quotation marks (' ').

CEMT PERFORM ENDAFFINITY

End an affinity owned by CICS.

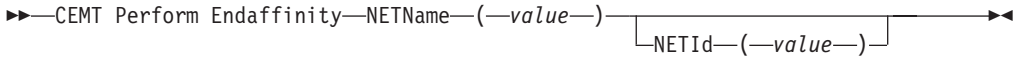
Description

Where CICS is a member of a VTAM generic resource group, PERFORM ENDAFFINITY 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 (as displayed by CEMT INQUIRE/SET CONNECTION) must be NORECOVDATA.

Generic resources and affinities are described in the *CICS Intercommunication Guide*.

Syntax

CEMT PERFORM ENDAFFINITY



Options

NETId (*value*)
specifies the 8-character name by which the network containing the connected 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.

NETName (*value*)
specifies the 8-character APPLID of the connected LU. If the connected LU is a member of a generic resource, you must specify its member name, not the generic resource name.

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 you entered the NETNAME or NETID incorrectly, or that you (or CICS) was wrong in supposing that an affinity existed.

CEMT PERFORM JVMPOOL

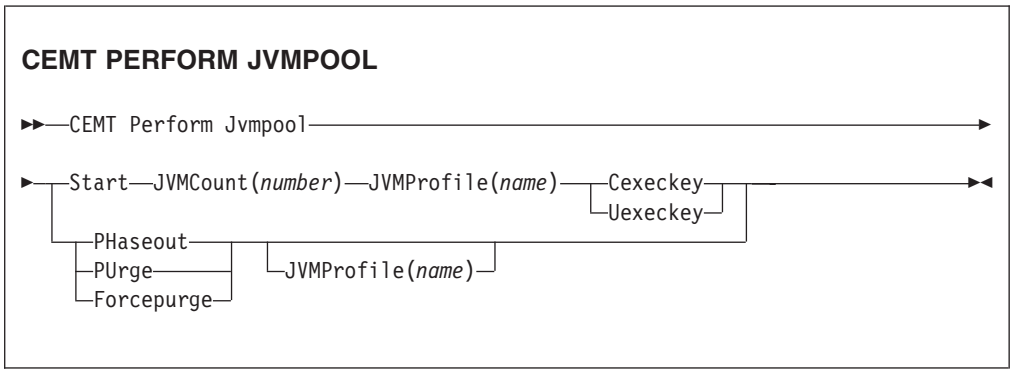
Start and terminate JVMs in the JVM pool.

Description

You can use the CEMT PERFORM JVMPOOL command to start JVMs with your chosen JVM profile and execution key. You can also use the command to terminate all or some of the JVMs in the pool, in order to implement changes to JVM profiles, or to add new application classes.

The CEMT PERFORM JVMPOOL TERMINATE command does not terminate the shared class cache and the master JVM. If the CICS region has a shared class cache, and you want to terminate it, use the CEMT PERFORM CLASSCACHE command to do this.

Syntax



Options

Cexeckey
Specifies that the JVMs to be started are to run in CICS key.

Forcepurge
All tasks using JVMs with the profile specified by the Jvmprofile option, or (if Jvmprofile is not specified) using all JVMs in the pool, are terminated by the SET TASK FORCEPURGE mechanism, and the JVMs are terminated.

Jvmcount(*number***)**
Specifies the number of JVMs to be started. If the number of JVMs you request, added to the number of JVMs that already exist, would mean exceeding the MAXJVMTCBS limit for the CICS region, CICS does not start any JVMs.

Jvmprofile(*name***)**
Specifies the 8-character name of a JVM profile. The name is case-sensitive, and you must enter it using the same combination of upper and lower case

characters that is present in the z/OS UNIX file name. If you need to enter the name of a JVM profile in mixed case when you are using the CEMT transaction, ensure that the terminal you use is correctly configured, with upper case translation suppressed.

Jvmprofile is required with Start, to specify the profile for the JVMs that are to be started.

Jvmprofile can optionally be specified with Phaseout, Purge, or Forcepurge, to mean that only JVMs with this profile should be terminated. Without the Jvmprofile option, these options act on all the JVMs in the pool.

Phaseout
JVMs with the profile specified by the Jvmprofile option, or (if Jvmprofile is not specified) all JVMs in the pool, are marked for deletion. The JVMs are actually deleted when they finish running their current Java program.

Purge
All tasks using JVMs with the profile specified by the Jvmprofile option, or (if Jvmprofile is not specified) using all JVMs in the pool, are terminated by the SET TASK PURGE mechanism, and the JVMs are terminated.

Start(*number***)**
Specifies a number of JVMs to be started. Jvmcount, Jvmprofile and either Cexeckey or Uexeckey are required with Start, to specify the number, profile and execution key for the JVMs. CICS starts the JVMs asynchronously.

You cannot specify Start when the JVM pool's status is set to Disabled.

Uexeckey
Specifies that the JVMs to be started are to run in user key. If the system initialization parameter STGPROT=NO is in effect, this is ignored, and the JVMs are started in CICS key.

CEMT PERFORM PIPELINE

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.

Description

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.

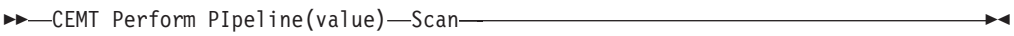
If, for any reason, CICS fails to install an individual Web service binding file, processing continues with the remaining files in the directory. When the scan completes, the PIPELINE is available for use with whichever of the binding files were installed successfully.

A failure to perform the requested action will be reported by returning a condition and RESP2 value from the command and, in certain cases, by writing messages to CSMT. The requested action may have been partially successful since a failure

does not cause the effects of the command to be backed out. Also, failure does not necessarily stop the command from continuing in order to increase the extent to which it was partially successful.

Syntax

CEMT PERFORM PIPELINE



Options

PIPELINE (*value***)**
specifies the 8-character PIPELINE name.

Scan
Scan the PIPELINE's Web service binding directory

CEMT PERFORM RESET

Synchronize the CICS date and time-of-day with the system date and time-of-day.

Description

To ensure that the correct local time is used by all CICS functions, **use the CEMT PERFORM RESET command immediately whenever you alter the system date or the time-of-day in the MVS TOD clock while a CICS region is running.** For example, when setting clocks forward or back an hour to adjust for Summer and Winter time, use this command to ensure that CICS immediately updates the correct local time.

Note: To ensure that your applications are receiving the correct time, always use **CEMT PERFORM RESET** whenever you alter the MVS TOD clock. CICS only obtains and stores the *local time offset*, at start up, and when the CEMT PERFORM RESET command executes.

Syntax

CEMT PERFORM RESET

CEMT PERFORM SECURITY

Rebuild the in-store RACF resource profiles. The RACF resource profiles are rebuilt by reissuing the RACLIST macro for the active class names.

Syntax

CEMT PERFORM SECURITY

If the external security manager is RACF 2.1 or an equivalent product, the PERFORM SECURITY command is not required to refresh the classes used for

resource checks. A message of NOT REQUIRED is displayed and no action is taken.

Options

Rebuild
is an optional keyword. It does not alter the action of the transaction.

CEMT PERFORM SHUTDOWN

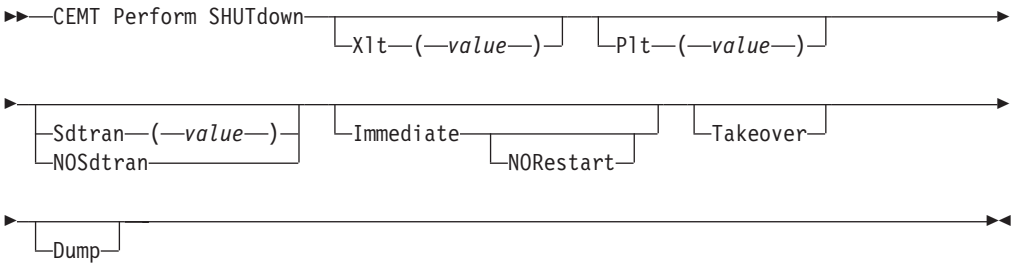
Shut down the CICS region.

Description

The shutdown can be either controlled or immediate. If a shutdown (either normal or immediate) hangs, a subsequent SHUTDOWN IMMEDIATE resumes the shutdown process.

Syntax

CEMT PERFORM SHUTDOWN



Options

Dump
A dynamic storage dump is produced upon completion of the termination process.

Immediate
specifies that the system is shut down immediately, terminating all active tasks and SNA sessions. If IMMEDIATE is not specified, all tasks are allowed to finish, and SNA sessions are allowed to terminate normally.

NORestart
specifies that, if the CICS region is registered to MVS/ESA Automatic Restart Manager (ARM), it is to be deregistered as part of shutdown and not restarted automatically.

NOSdtran
No shutdown assist transaction is to be run at CICS shutdown.

The default shutdown transaction, if neither SDTRAN nor NOSDTRAN are specified, is CESD.

Plt (*value*)
specifies the 2-character suffix of a program list table (PLT) to be used for the shutdown.

Sdtran
The 4-character identifier of the shutdown assist transaction

Takeover
specifies that this CICS region should be closed down, and the alternate CICS region is to take over. This is valid only if the XRF=YES system initialization parameter was specified for CICS startup.

Xlt (*value***)**
specifies the 2-character suffix of a transaction list table (XLT) to be used for the shutdown.

Transactions listed in the specified XLT can be initiated after the SHUTDOWN request, and before the system quiesces. The default XLT should contain an entry for the master terminal transaction. This ensures that you can issue an immediate shutdown if, for example, a task is found to be suspended after an ordinary (nonimmediate) shutdown has been requested.

Table 5 shows the effect of the TAKEOVER and IMMEDIATE options on the active CICS system.

Table 5. Effect of the TAKEOVER and IMMEDIATE options on the active CICS system

	CEMT PERFORM SHUTDOWN		
	(normal)	***TAKEOVER***	***IMMEDIATE***
Tasks	finish	finish	terminate
SNA session (XRF)	unbind	leave	leave
SNA session (non-XRF)	unbind	unbind	unbind

Table 6 shows the effect of the TAKEOVER and IMMEDIATE options on the alternate CICS system.

Table 6. Effect of the TAKEOVER and IMMEDIATE options on the alternate CICS system

	CEMT PERFORM SHUTDOWN		
	(normal)	***TAKEOVER***	***IMMEDIATE***
CICS alternate	shuts down	takes over	takes over

For further information on XRF, see the *CICS/ESA 3.3 XRF Guide*.

CEMT PERFORM STATISTICS

Write the statistics for all, or a named resource type immediately to the SMF data set, rather than wait for the current statistics-gathering interval to expire.

Description

The recording is made irrespective of the ON or OFF setting of statistics. See also “CEMT INQUIRE STATISTICS” on page 265 and “CEMT SET STATISTICS” on page 435 for a description of the CEMT INQUIRE and SET STATISTICS commands.

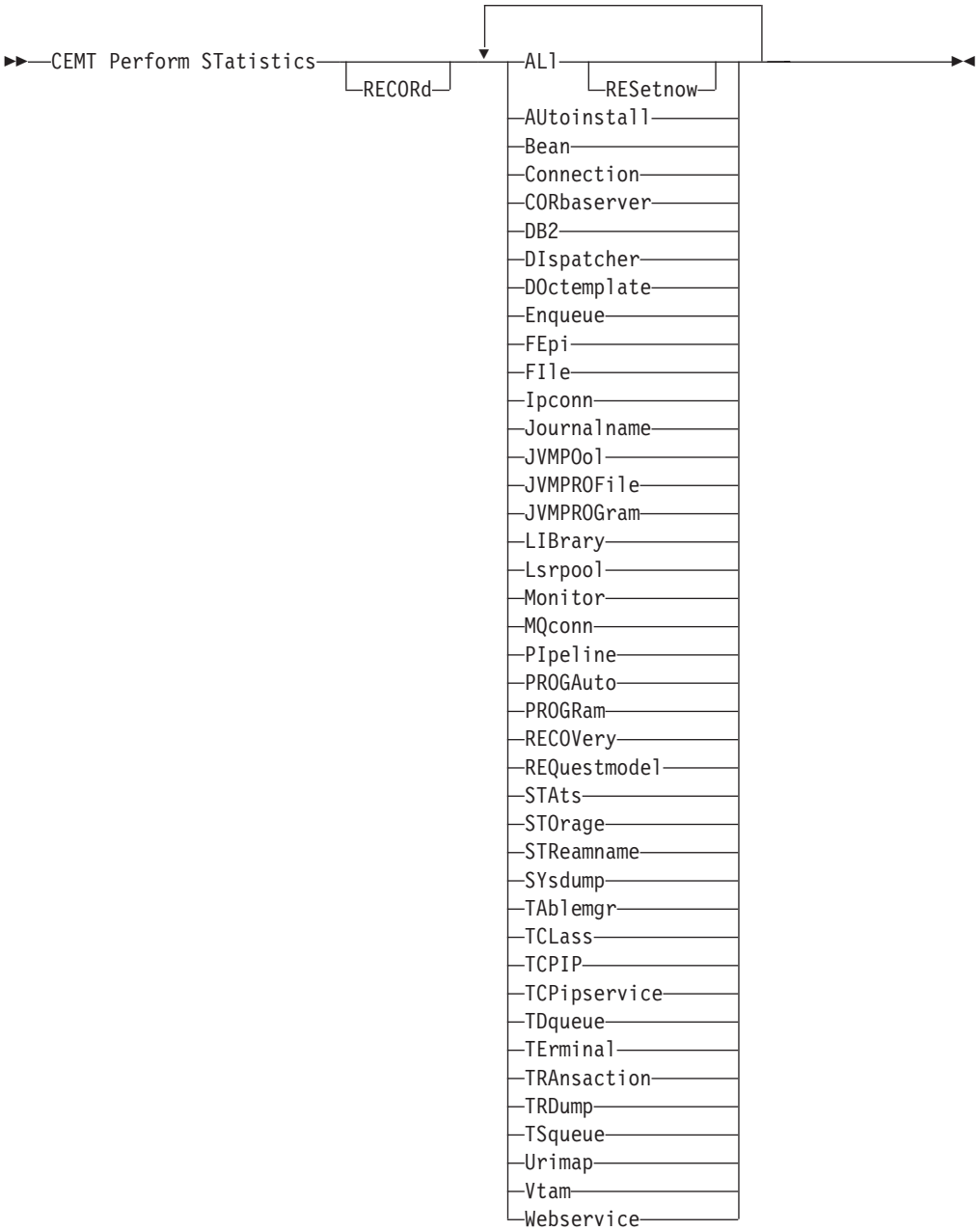
You can request either statistics for all resources (ALL), or statistics for individual resources in various combinations by selecting one or more keywords from the list.

The message ‘STATS MISSING’ appears when the resource type is either not functioning or not available. As much data is recorded as possible.

See the *CICS Operations and Utilities Guide* for information about printing statistics using the CICS-supplied utility program DFHSTUP, and for information about the reports produced by DFHSTUP.

Syntax

CEMT PERFORM STATISTICS



Options

ALl

Statistics for all resources are to be written immediately to the SMF data set.

ALL RESETNOW resets the counters for the collection of statistics. RESETNOW only works if the ALL operand is set, that is, ALL RESETNOW.

AUtoinstall
Autoinstall statistics are to be written immediately to the SMF data set.

Bean
Enterprise bean statistics are to be written immediately to the SMF data set.

Connection
Connection statistics are to be written immediately to the SMF data set.

CORbaserver
CorbaServer statistics are to be written immediately to the SMF data set.

DB2
CICS DB2 statistics are to be written immediately to the SMF data set.

DIspatcher
Dispatcher statistics are to be written immediately to the SMF data set.

DOctemplate
Document template statistics are to be written immediately to the SMF data set.

Enqueue
Enqueue statistics are to be written immediately to the SMF data set.

FEpi
FEPI statistics are to be written immediately to the SMF data set.

FIle
File control statistics are to be written immediately to the SMF data set.

Ipconn
IPCONN statistics are to be written immediately to the SMF data set.

Journalname
CICS journal statistics are to be written immediately to the SMF data set.

Note: The JOURNALNUM resource name is obsolete, and replaced by the JOURNALNAME resource name.

JVMPOol
JVM pool statistics are to be written immediately to the SMF data set.

JVMPROFile
JVM profile statistics are to be written immediately to the SMF data set.

JVMPROGram
JVM program statistics are to be written immediately to the SMF data set.

LIBrary
LIBRARY statistics are to be written immediately to the SMF data set.

Lsrpool
LSRPOOL statistics are to be written immediately to the SMF data set.

Monitor
Monitoring statistics are to be written immediately to the SMF data set.

MQconn
WebSphere MQ Connection statistics are to be written immediately to the SMF data set.

PIpeline
PIPELINE statistics are to be written immediately to the SMF data set.

PROGAuto
Autoinstalled program statistics are to be written immediately to the SMF data set.

PROGRam
Program statistics are to be written immediately to the SMF data set.

RECORd
is an optional keyword. It does not alter the action of the transaction.

RECOVery
Recovery statistics are to be written immediately to the SMF data set.

REQuestmodel
Requestmodel statistics are to be written immediately to the SMF data set.

RESetnow
The statistics counters are to be reset.

STAts
Statistics domain statistics are to be written immediately to the SMF data set.

STOrage
Storage manager statistics are to be written immediately to the SMF data set.

STReamname
Log stream statistics are to be written immediately to the SMF data set.

SYsdump
System dump statistics are to be written immediately to the SMF data set.

TAblemgr
Table manager statistics are to be written immediately to the SMF data set.

TCLass
Transaction class statistics are to be written immediately to the SMF data set.

TCPIP
Request global statistics for IP sockets.

TCPipservice
TCP/IP service statistics are to be written immediately to the SMF data set.

TDqueue
Transient data statistics are to be written immediately to the SMF data set.

TErminal
Terminal statistics are to be written immediately to the SMF data set.

TRAnsaction
Transaction statistics are to be written immediately to the SMF data set.

TRDump
Transaction dump statistics are to be written immediately to the SMF data set.

TSqueue
Temporary storage statistics are to be written immediately to the SMF data set.

Urimap
URIMAP statistics are to be written immediately to the SMF data set.

Vtam
VTAM statistics are to be written immediately to the SMF data set.

Webservice
WEBSERVICE statistics are to be written immediately to the SMF data set.

CEMT SET commands

The SET command changes the status of one or more named instances of a resource (such as a terminal), a particular subgroup of resources (such as a class of terminals), or all resources of a particular type (such as all terminals).

The SET command makes the changes that you specify, and displays the new status of the resources. No changes are made if there are syntax errors in the SET command.

To change the attributes of a resource, enter SET (or a suitable abbreviation) on the command line. The keywords you can use with CEMT SET are described in the rest of this section.

On every SET command, an argument, a generic argument, or ALL must be specified. For example:

```
SET TERMINAL(TRM1) INSERVICE
```

sets terminal TRM1 in service.

```
SET TERMINAL(TRM1 TRM2 TRM3) INSERVICE
```

or

```
SET TERMINAL(TRM1,TRM2,TRM3) INSERVICE
```

sets terminals TRM1, TRM2, and TRM3 in service.

```
SET TERMINAL(TRM*) INSERVICE
```

sets all terminals with identifiers starting with TRM in service.

```
SET TERMINAL ALL INSERVICE
```

sets all terminals in service.

However,

```
SET TERMINAL INSERVICE
```

produces an error.

CEMT SET AUTOINSTALL

Change autoinstall values.

Description

The CEMT SET AUTOINSTALL command allows you to change some of the values that control the automatic installation (autoinstall) of VTAM terminals, virtual terminals including bridge facilities used by the 3270 bridge mechanism, and MVS consoles. For information about autoinstall, see the *CICS Resource Definition Guide*.

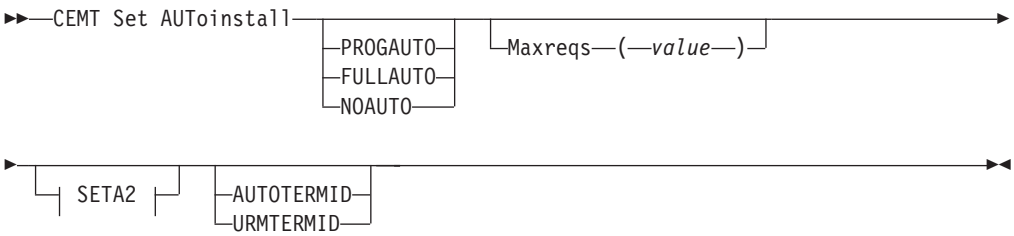
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

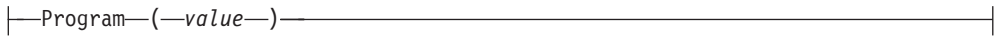
- Type CEMT SET AUTOINSTALL (the minimum abbreviation is CEMT S AUTO). You get a display that lists the current status, similar to that obtained by CEMT INQUIRE AUTOINSTALL. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type CEMT SET AUTOINSTALL (CEMT S AUTO) followed by one or more attribute settings that you wish to change. For example, CEMT S AUT M(NEWVALUE P(NEWNAME) resets the values for maximum number of requests for a new program.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET AUTOINSTALL



SETA2



Options

Aibridge
Specifies whether the autoinstall user replaceable module (URM) is to be called for bridge facilities. The values are:

AUTOTERMID
Bridge facilities are to be defined automatically by CICS. The autoinstall URM is not to be called.

URMTERMID
The autoinstall URM is to be called.

Consoles
specifies whether CICS is to autoinstall an MVS console device 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 (*value*)
specifies the largest number of autoinstall requests that are allowed to queue at one time, in the range 0–999.

You can prevent more terminals from logging on through autoinstall by setting this value to 0. This allows autoinstalled entries for terminals currently logged on to be deleted by the autoinstall program when they log off.

Program (*value*)
specifies the name of the user program that is to control the autoinstall process. A default program is supplied.

CEMT SET AUXTRACE

Change auxiliary tracing options.

Description

For more information about traces, see the *CICS Problem Determination Guide*. See also the description of the CETR transaction, Chapter 20, “CETR - trace control,” on page 479.

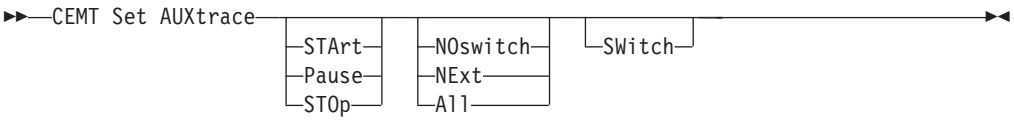
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET AUXTRACE` (the minimum abbreviation is `CEMT S AUX`). You get a display that lists the current status, similar to that obtained by CEMT INQUIRE AUXTRACE. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET AUXTRACE` (`CEMT S AUX`) followed by one or more attribute settings that you wish to change. For example, `cemt s aux sto a` stops auxiliary tracing and causes automatic switching to occur as necessary.

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET AUXTRACE



Options

All
Specifies that automatic switching between the two auxiliary trace data sets is to occur as necessary until the end of this CICS session, without the need for operator intervention.

NExt
Automatic switching of the auxiliary trace data set is to occur when the current trace data set is full, but not subsequently.

NOswitch
Switching between the auxiliary trace data sets is not to occur without operator intervention.

Pause
Auxiliary tracing is to stop, but the data set is to remain open. A subsequent START request causes trace entries to be written immediately following those that were written before the PAUSE request.

STArt
CICS is to start auxiliary tracing and open the auxiliary trace data set if it is currently closed.

STOp
CICS is to stop auxiliary tracing and close the auxiliary trace data set. A subsequent START request causes new trace entries to be written at the start of the data set, thereby overwriting the trace entries that were written before the STOP request.

SWitch
specifies that an immediate switch from the current auxiliary trace data set is to be made to the alternate trace data set.

CEMT SET BRFACILITY

Release virtual terminals (bridge facilities) used by the 3270 bridge mechanism.

Description

CEMT SET BRFACILITY 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.

Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET BRFACILITY` (the minimum abbreviation is `CEMT S BR`) with (*value*) or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE BRFACILITY. You can then tab to the highlighted field and overtype it with the required value.
- Type `CEMT SET BRFACILITY` (`CEMT S BR`) with value or ALL, followed by one or more attribute settings that you wish to change. For example, `cemt s br(`*value*`) rel` will release the specified bridge facility.

To obtain a syntax prompt, enter `?CEMT S BR`.

CEMT SET BRFACILITY

Options

(*value***)**
is an 8–byte bridge facility token.

All
Any changes you request are made to all resources of the specified type that you are authorized to access.

RELeased
specifies that the bridge facility should be marked for deletion.

CEMT SET CLASSCACHE

Set the status of autostart for the shared class cache.

Description

On an initial or cold start of CICS, the status of autostart is taken from the system initialization parameter JVMCCSTART. A JVMCCSTART value of YES or AUTO gives an autostart status of ENABLED, and a JVMCCSTART value of NO gives an autostart status of DISABLED.

When you change the status of autostart for the shared class cache while CICS is running, subsequent warm or emergency CICS restarts use the changed setting, unless the JVMCCSTART system initialization parameter is specified as an override at startup, in which case the behavior from the system initialization parameter is used. This does not affect the shared class cache provided by the IBM SDK for z/OS, V5 because it persists across warm or emergency restarts, but it does affect the Version 1.4.2 shared class cache. If the system is initial or cold started, the status of autostart is always taken from the JVMCCSTART system initialization parameter.

Syntax

CEMT SET CLASSCACHE

Options

Disabled
Sets the status of autostart for the shared class cache to Disabled.

When autostart is disabled, if the shared class cache has been stopped or has not yet been started on this CICS execution, an explicit PERFORM CLASSCACHE command is required to start it. If the status of the shared class cache is STOPPED and autostart is disabled, and CICS receives a request to run a Java application in a JVM whose profile requires the use of the shared class cache, the application cannot run.

Enabled
Sets the status of autostart for the shared class cache to Enabled.

When autostart is 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.

CEMT SET CONNECTION

Change the attributes of an MRO or ISC over SNA connection.

Description

CEMT SET CONNECTION allows you to change some of the attributes that define an MRO or ISC over SNA connection.

Note: See also "CEMT SET IPCONN" on page 413. The CEMT SET IPCONN command is used to change the attributes of IPIC connections (also known as "*IPCONNs*").

For information about the different kinds of intercommunication connections, see the *CICS Intercommunication Guide*.

When a connection has been defined as INDIRECT, no information is given unless the direct connection to which it refers is installed.

If you want to install a new connection definition when one is already installed, you must set the connection OUTSERVICE and, if it is VTAM-connected, release it before using the CEDA INSTALL commands for your new definition.

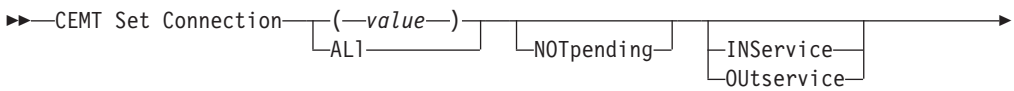
Syntax

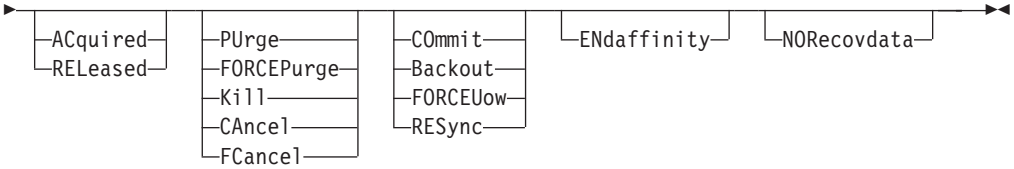
Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET CONNECTION` (the minimum abbreviation is `CEMT S C`) followed by one or more connection identifiers or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE CONNECTION. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET CONNECTION` (`CEMT S C`) followed by one or more connection identifiers or ALL, followed in turn by one or more attribute settings that you wish to change. For example, `cemt s c al i` resets the values for all connections to make them available for use (inservice).

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET CONNECTION





Options

(*value***)**
One or more names (1–4 characters) defined for an interregion communication (IRC) or an intersystem communication (ISC) connection.

ACquired (*appc only***)**
specifies whether CICS is to acquire a session with the logical unit represented by the CONNECTION name. To get more detailed information about the availability status of the connection elements, use the CEMT INQUIRE MODENAME command. A connection cannot be both ACQUIRED and OUTSERVICE.

ALl
Any changes you request are made to all resources of the specified type that you are authorized to access.

Backout (APPC parallel-session, CICS-to-CICS MRO, and LU61 only)
All UOWs shunted due to the failure of this connection are to be backed out. The normal resynchronization process is to be partially overridden: decisions are taken for any units of work that are in-doubt due to a failure of the connection; but the decisions are recorded and any data inconsistencies are reported when the connection is next acquired.

CAncel
Automatic initiate descriptors (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 be canceled. In addition, the following CICS system AIDs are not purged unless FCANCEL is specified.

Table 7. System AIDs requiring FCANCEL to remove them

Description	Tranid
Remote delete AIDs	
Remote scheduler AIDs	CRSR
LU6.2 service manager 1 AIDs	CLS1
LU6.2 service manager 3 AIDs	CLS3
Remote scheduler PURGE AIDs	CRSQ
Resource manager resync AIDs	CRSY
Autoinstalled terminal delete AIDs	CATD
Autoinstalled terminal restart AIDs	CATR

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, the remote AID is canceled asynchronously.

Message DFHTF0100 is written to CSMT to indicate how many AIDs have been deleted for the connection and how many remain.

An 'AIDS CANCELED' message appears on the CEMT panel whenever AIDs are deleted using the CANCEL option of the CEMT SET CONNECTION command.

COmmit (APPC parallel-session, CICS-to-CICS MRO, and LU61 only)
All UOWs shunted due to the failure of this connection are to be committed. The normal resynchronization process is to be partially overridden: decisions are taken for any units of work that are in-doubt due to a failure of the connection; but the decisions are recorded and any data inconsistencies are reported when the connection is next acquired.

ENdaffinity (*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.

Note:

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 you must end explicitly, CICS issues message DFHZC0177. This message gives the NETNAME 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. A response of 'NETID 0 USE PRFRM' means that you cannot use SET CONNECTION to end the affinity because the value of NETID in the installed connection is 0. (NETID is the name by which the network containing the connected LU is known to VTAM.) To end the affinity, you must use the PERFORM ENDAFFINITY command, on which you must specify the correct NETID.

   This response can also result from repeatedly hitting the ENTER key after the affinity has been ended.
4. Generic resources and affinities are described in the *CICS Intercommunication Guide*.

FCancel
All AIDs, including system AIDs, queuing for the specified connection are to be canceled. See Table 7 on page 376 for a list of those system AIDS that require FCANCEL to remove them. This can lead to unpredictable results and should be used only in exceptional circumstances.

Note: FCANCEL does not remove transient data AIDs with an associated triggered task. These aids may be removed by purging the associated task.

An 'AIDS CANCELED' message appears on the CEMT panel whenever AIDs are deleted using the FCANCEL option of the CEMT SET CONNECTION command.

FORCEPurge (*vtam only***)**
All transactions running on sessions on the connected system are immediately terminated abnormally. This can lead to unpredictable results and should be used only in exceptional circumstances.

In some extreme cases (for example, if an error occurs during backout processing), CICS might terminate abnormally.

FORCEPURGE replaces PURGE FORCE which is retained only for compatibility purposes. You should use FORCEPURGE in new applications.

FORCEUow (APPC parallel-session, CICS-to-CICS MRO, and LU61 only)
All UOWs shunted due to the failure of this connection are to be forced to back out or commit, as specified in the ACTION option of the TRANSACTION definition. The normal resynchronization process is to be partially overridden: decisions are taken for any units of work that are in-doubt due to a failure of the connection; but the decisions are recorded and any data inconsistencies are reported when the connection is next acquired.

INService
The system is in service; that is, it is available for use.

For an MRO connection, all sessions are placed in service and the following occurs:

- If both the issuing system and system “sysid” have IRC open, and the issuing system has status INSERVICE within system “sysid”, connections are established and made available between the two systems.
- Otherwise, the status of the system (and the status of the underlying sessions) is set INSERVICE, so that when both systems have IRC open and are INSERVICE with respect to each other, connections are established. Note that INQUIRE indicates that systems (and their underlying sessions) are INSERVICE even though no connections can be used.
- The status of the underlying sessions for a system is always the same as that for the system itself.

For an ISC system, the following occurs:

- LU6.1 system (that is, links to “sysid” are through intersystem communication) - all underlying sessions are placed in service.
- APPC system—causes the SNASVCMG sessions to be placed in service thereby enabling the connection subsequently to be established.

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.

NORecovdata (*appc only*)
specifies that the normal resynchronization process is to be overridden. NORECOVDATA forces 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.

Note: 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: If you specify NORECOVDATA you cannot specify COMMIT, BACKOUT, FORCEUOW, RESYNC, or NOTPENDING.

NOTpending (*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 TS for z/OS, Version 3.2 partner that has performed an initial start
- A connection to a pre- CICS TS for z/OS, Version 3.2 partner that has performed a cold start.

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 TS for z/OS, Version 3.2 partner, a lognames mismatch indicates that the partner has performed an initial start. For a pre-CICS TS for z/OS, Version 3.2 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.

Note:

1. If you specify NOTPENDING you cannot specify COMMIT, BACKOUT, FORCEUOW, RESYNC, or NORECOVDATA. For advice on which option to use, see CEMT SET CONNECTION - notes on RES ync.
2. NOTPENDING has no effect on MRO connections to pre-CICS TS for z/OS, Version 3.2 systems. However, the cold start of a pre-CICS TS for z/OS, Version 3.2 MRO partner causes the SET CONNECTION NOTPENDING function to be performed automatically, session by session.

The exchange lognames function and the resynchronization function are described in the *CICS Intercommunication Guide* and the *Systems Network Architecture—LU6.2 Reference: Peer Protocols* manual.

OUtservice
The system is out of service; that is, it is not available for use.

For an MRO 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 the 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 ISC system, the following occurs:

- LU6.1 system—all underlying sessions owned by the system 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.
- APPC system—this option is valid only if the system is released. All sessions owned by the ISC system are then out of service and released.

PUrge (*vtam only and irc only*)
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.

RELeased (*appc only*)
specifies whether CICS is to release a session with the logical unit represented by the CONNECTION name. To get more detailed information about the availability status of the connection elements, use the CEMT INQUIRE MODENAME command.

RESync (*mro to CICS TS for z/OS, Version 3.2 and later systems and appc only*) **(APPC parallel-session, CICS-to-CICS MRO, and LU61 only)**
Any UOWs shunted due to 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. The normal resynchronization process is to be partially overridden: decisions are taken for any units of work that are in-doubt due to a failure of the connection; but the decisions are recorded and any data inconsistencies are reported when the connection is next acquired.

Note:

1. The COMMIT, BACKOUT, FORCEUOW and RESYNC operations are synchronous with setting the state of the UOW; that is, an INQUIRE UOW following SET CONNECTION BACKOUT, COMMIT, FORCEUOW, or RESYNC returns the new UOW states.
2. Specifying one of these options unshunts all units of work that have failed due to a failure of the connection. Before issuing SET CONNECTION FORCEUOW, 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 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.

3. You can specify only one of the BACKOUT, COMMIT, FORCEUOW, RESYNC, NOTPENDING, and NORECOVDATA options. SET CONNECTION NORECOVDATA should be used only in exceptional circumstances.
4. 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.
5. The BACKOUT, COMMIT, FORCEUOW, or RESYNC options of SET CONNECTION and SET UOW do not clear resync information. If you want to do this, you must use SET CONNECTION NOTPENDING or SET CONNECTION NORECOVDATA.
6. You can issue BACKOUT, COMMIT, FORCEUOW, or RESYNC commands *before* issuing SET CONNECTION NOTPENDING or SET CONNECTION NORECOVDATA.

CEMT SET CORBASERVER

Enable or disable a CorbaServer, turn its automatic publishing feature on or off, and set the session bean timeout value.

Description

If the optional attributes are omitted, the command has no effect.

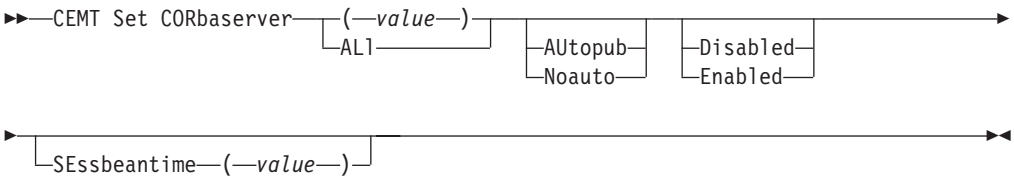
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET CORBASERVER` (the minimum abbreviation is `CEMT S COR`) followed by one or more CORBASERVER identifiers or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE CORBASERVER. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET CORBASERVER` (`CEMT S COR`) followed by one or more CORBASERVER identifiers or ALL, followed in turn by one or more attribute settings that you wish to change. For example, `cemt s c al i` resets the values for all CORBASERVERs to make them available for use (inservice).

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET CORBSERVER



Options

ALl
Any changes you request are made to all resources of the specified type that you are authorized to access.

(*value***)**
specifies the 1–4 character name of the CorbaServer.

AUtopub
specifies that the contents of a deployed JAR file should be automatically published to the namespace when the DJAR definition is successfully installed into this CorbaServer.

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.

Disabled
specifies that the CorbaServer is to be disabled.

In a CICS region containing multiple CorbaServers, putting one of the CorbaServers into disabled state allows you to apply maintenance to it without bringing down CICS; the other CorbaServers remain available.

Similarly, in a logical EJB/CORBA server containing multiple AORs (each containing multiple CorbaServers), putting a CorbaServer on one of the AORs into disabled state allows you to apply maintenance to the CorbaServer without needing to remove the AOR from the routing set.

You can disable a CorbaServer that is in any state except ENABLING or DISCARDING. When you do so, any currently-executing methods are allowed to continue to completion. Until all its objects have been destroyed or passivated, the CorbaServer enters DISABLING state. Note that:

- Any instance of an enterprise bean or CORBA stateless object that is not participating in an OTS transaction is destroyed or passivated at the end of the currently-executing method.
- Any instance of an enterprise bean or CORBA stateless object that *is* participating in an OTS transaction is not destroyed or passivated until the end of the OTS transaction; typically, any future method calls against this instance (within the scope of the OTS transaction) will succeed. At the end of the OTS transaction, the instance is destroyed or passivated.

If the workload manager directs new work to an AOR in which the target CorbaServer is in DISABLING or DISABLED state, it is informed that the CorbaServer is disabled.

Note: By “new work” we mean:

- Requests for methods that will run under a new OTS transaction. (By a “new” OTS transaction we mean an OTS transaction in which none of the AORs in the logical server is already participating, 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
specifies that the CorbaServer is to be enabled. You can enable a CorbaServer that is in any state except DISABLING or DISCARDING.

Noauto
specifies that the contents of a deployed JAR file should *not* be automatically published to the namespace when the DJAR definition is successfully installed into this CorbaServer.

SEssbeantime (*value*)
specifies the elapsed time period in minutes, in the range 0 through 143999, of inactivity after which a session bean may be discarded. A value of 0 indicates that beans are not timed out.

CEMT SET DB2CONN

Set attributes of the CICS DB2 connection including the pool and command thread attributes.

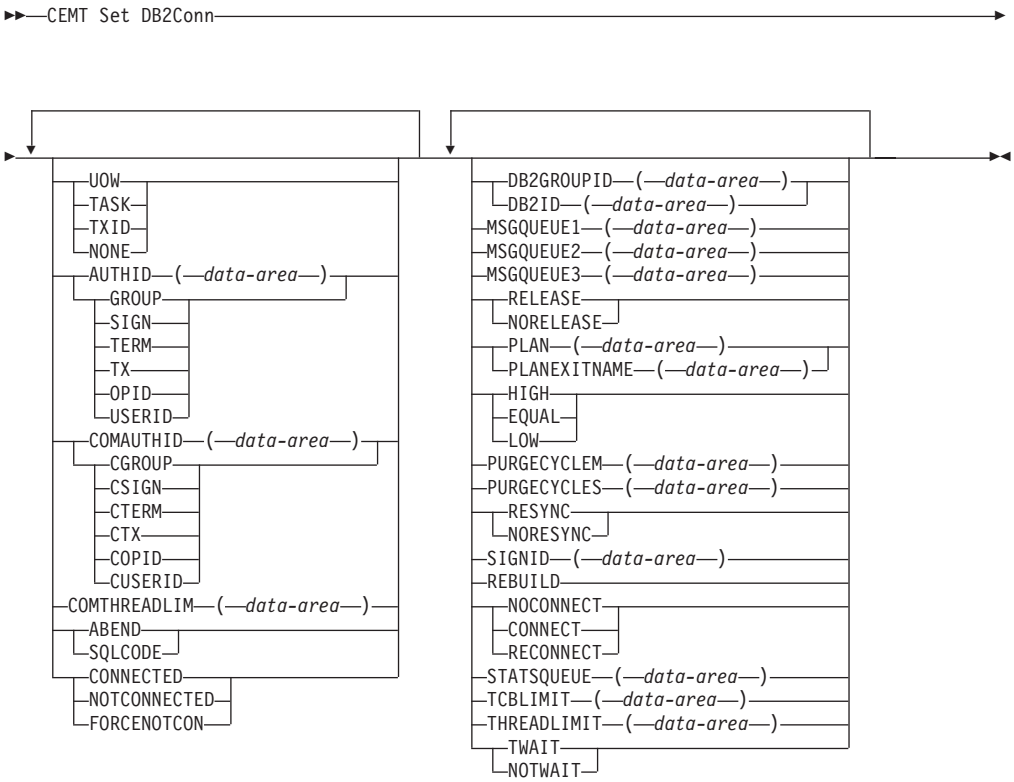
Syntax

Press the Clear key to clear the screen. You can start this transaction in two ways:

- Type `CEMT SET DB2CONN` (the minimum abbreviation is `CEMT S DB2C`). The resulting display lists the current status, similar to that obtained by using the CEMT INQUIRE DB2CONN command. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET DB2CONN` (`CEMT S DB2C`) followed by one or more attribute settings that you wish to change.

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET DB2CONN



Options

Accountrec

Specifies the minimum amount of DB2 accounting required for transactions requiring pool threads. The specified minimum can be exceeded, as described in the following options. CVDA values are as follows:

UOW The CICS DB2 attachment facility causes an accounting record to be produced for each UOW, assuming that the thread has been released at the end of the UOW.

TASK The CICS DB2 attachment facility causes a minimum of one accounting record for each CICS task to be produced.

A transaction that contains multiple UOWs might use a different thread for each of its UOWs (assuming the thread is released at syncpoint). The result might be an accounting record produced for each UOW.

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 that contains multiple UOWs will use a different thread for each UOW (assuming the thread is released at syncpoint). In this case, an accounting record might be produced for each UOW.

NONE

No accounting records are required for transactions using pool threads.

DB2 does, however, produce at least one accounting record for each thread when the thread is terminated. Additionally, authorization changes cause accounting records to be produced.

Authid

Specifies what ID to use for security checking for pool threads. If AUTHID is specified, AUTHTYPE is cleared.

Authtype

Specifies the type of ID that can be used for pool threads. If AUTHType is specified, AUTHID is cleared. CVDA values are as follows:

GROUP

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

IDs passed to DB2	How DB2 interprets values
CICS sign-on user ID (USERID)	Represents the primary DB2 authorization ID.
RACF connected group name	If the RACF list of group options is not active, 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.

To use the GROUP option the CICS system must have SEC=YES specified in the CICS system initialization table (SIT).

If no RACF group ID is available for this USERID, an 8-character field of blanks is passed to DB2 as the group ID.

SIGN Specifies the SIGNID parameter of the DB2CONN should be used as the resource authorization ID.

TERM

Specifies the terminal identification (four characters padded to eight) as an authorization ID. An authorization ID cannot be obtained in this manner if a terminal is not connected with the transaction.

If a transaction is started (using a CICS command) and has no terminal associated with it, AUTHTYPE(TERM) should not be used.

TX Specifies the transaction identification (four characters padded to eight) as the authorization ID.

OPID The operator identification associated with the userid that is associated with the CICS transaction is used as the authorization ID (three characters padded to eight).

USERID

The 8-character 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).

Comauthid

Specifies which ID to use for security checking when using command threads. If COMAUTHid is specified, COMAUTHType is cleared.

Comauthtype

Specifies the type of ID that can be used for security checking when using command threads. If COMAUTHType is specified, COMAUTHid is cleared. CVDA values are as follows:

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.

IDs passed to DB2	How DB2 interprets values
CICS sign-on user ID (USERID)	Represents the primary DB2 authorization ID.
RACF connected group name	If the RACF list of group options is not active, 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.

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 that 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 a value when CICS is not connected to DB2 because the adapter is in standby mode. The value shows how this information is passed back to an application that has issued a SQL request. CVDA values are as follows:

ABEND
The application is abended with abend code AEY9.

SQLCODE
The application receives a -923 SQLCODE.

Connectst
Sets the status of the CICS DB2 connection; that is, to start or to stop the CICS DB2 connection. CVDA values are as follows:

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
CEMT initiates a quiesce stop of the CICS DB2 connection and implicitly uses the NOWAIT option, meaning that control returns immediately to CEMT before the quiesce is complete.

FORCENOTCON
FORCENOTCON 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. With DB2 Version 10, the name can be a subgroup name identifying a subset of the data-sharing group.

DB2GROUPID can only be changed when CICS is not connected to a DB2 system. Specifying a DB2GROUPID causes the DB2ID in the installed DB2CONN definition to be blanked out. If an individual subsystem's DB2ID is specified in a CEMT or EXEC CICS SET DB2CONN command, or in a DSNC STRT command, this overrides any DB2GROUPID that is set in the installed DB2CONN definition. The DB2GROUPID is blanked out, and 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 as follows:

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 is cleared.

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 will be determined using the new rules. If PLANExitname is specified, PLAN is cleared.

Priority
Specifies the priority of the pool thread TCBs relative to the CICS main TCB (QR TCB). The thread TCBs are CICS open L8 TCBs. CVDA values are as follows:

HIGH Thread TCBs have a higher priority than the CICS QR TCB.

EQUAL
Thread TCBs have equal priority with the CICS QR TCB.

LOW Thread TCBs have a lower priority than the CICS QR TCB.

PURGECYCLEM
Specifies the number of minutes (in the range 00 - 59) in the protected thread purge cycle time. Use this parameter in conjunction with PURGECYCLES.

PURGECYCLES
Specifies the number of seconds (in the range 00 - 59) in the protected thread purge cycle time. Use this parameter in conjunction with PURGECYCLEM. The minimum and default protected thread purge cycle time is 30 seconds.

A protected thread is not terminated immediately when it is released. It is terminated only after two completed purge cycles, if it has not been reused in the meantime. For example, if the protected thread purge cycle is set to 30 seconds, a protected thread is purged 30 - 60 seconds after that thread is released. An unprotected thread is terminated when it is released (at sync point or end of task) if there are no other transactions waiting for a thread on that DB2ENTRY resource.

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 that 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 as follows:

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
Specifies the security. CVDA value is as follows:

REBUILD
Specifies that the CICS DB2 attachment should force all existing threads to sign-on 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 as follows:

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 intialization parameter MAXOPENTCBS). The TCBLIMIT attribute of the DB2CONN definition governs how many of the open TCBs can be used to access DB2 — that is, how many of them can identify to DB2 and create a connection into DB2.

THREADLimit
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, message DFHDB2011 when THREADWAIT=NO is coded and the number of pool threads is exceeded. CVDA values are as follows:

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 abend code AD3T or AD2P.

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.

When the SET DB2CONN command is specified, all parameters except the DB2ID (the connected subsystem) can be set when the CICS DB2 attachment is active. *DB2ID can be changed only by stopping and restarting the attachment*.

If you change the PLAN and PLANEXITNAME while there are active transactions for that entry, or the pool, the next time the transaction releases the thread, the plan or exit is determined using the new rules.

CEMT SET DB2ENTRY

Sets the attributes of a particular DB2ENTRY used to define resources for a specific transaction or a group of transactions when accessing DB2.

Description

The entry is identified by the name it was identified with in CEDA. For RCTs migrated to the CSD, the name is the name of the first transaction on the DSNCRCT TYPE=ENTRY statement.

All parameters on SET DB2ENTRY can be set while the CICS DB2 attachment is active and the transactions are active.

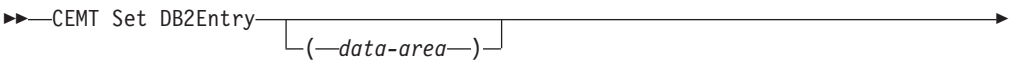
Syntax

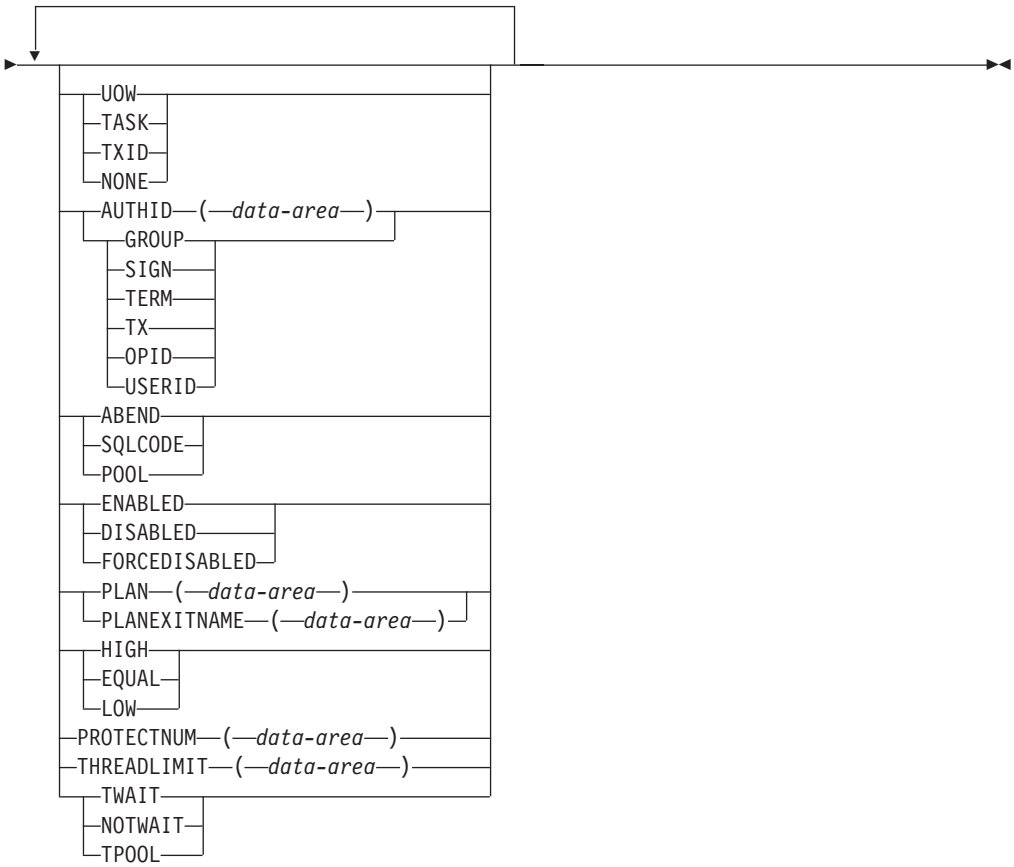
Press the Clear key to clear the screen. There are two ways of commencing this transaction

- Type `CEMT SET DB2ENTRY` (the minimum abbreviation is `CEMT S DB2E`) with either a value corresponding to a DB2 entry name or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE DB2ENTRY. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET DB2ENTRY` (`CEMT S DB2E`) with either the name of an entry or ALL, followed by one or more attribute settings that you wish to change. For example, `cemt s db2e all ena` enables all DB2ENTRYs.

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET DB2ENTRY





Options

Accountrec

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

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.

TASK The CICS DB2 attachment facility causes a minimum of one accounting record to be produced for each CICS task.

A transaction containing multiple UOWs can use a different thread for each of its UOWs (assuming the thread is released at syncpoint). The result can be an accounting record produced for each UOW.

TXID The CICS DB2 attachment facility produces at least one accounting record to be produced 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 st syncpoint, a transaction containing multiple UOWs can use a different thread for each UOW. The result can be that an accounting record is produced per UOW.

NONE

No accounting records are required for transactions using threads from this DB2ENTRY.

DB2 does, however, produce at least one accounting record for each thread when the thread is terminated. Additionally, authorization changes cause accounting records to be produced.

AUTHID
specifies the id to be used for security checking when using this DB2ENTRY. If AUTHId is specified, AUTHTYPE is cleared.

Authtype
returns the type of id that can be used for security checking when using this DB2ENTRY. If AUTHTYpe is specified, AUTHid is cleared. CVDA values are:

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

IDs passed to DB2	How DB2 interprets values
CICS sign-on user ID (USERID)	Represents the primary DB2 authorization ID.
RACF connected group name	If the RACF list of group options is not active, 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.

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).

Disabledact
specifies what CICS is to do with new transactions that access a DB2ENTRY when it has been disabled or is 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 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.

Enabledstatus
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.

FORCEDISABLED
All tasks using the DB2ENTRY and those queued against the DB2ENTRY are forcepurged. The DB2ENTRY is then disabled.

PLAN
specifies the name of the plan to be used for this DB2ENTRY.

If PLAN is specified, PLANEXITNAME is cleared.

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 is cleared.

Priority
specifies the priority of the thread TCBs for this DB2ENTRY relative to the CICS main TCB (QR TCB). The thread TCBs are CICS open L8 TCBs. CVDA values are:

HIGH Thread TCBs have a higher priority than the CICS QR TCB.

EQUAL
Thread TCBs have equal priority with the CICS QR TCB.

LOW Thread TCBs have a lower priority than the CICS QR TCB.

PROTECTNUM
specifies the maximum number of protected threads for this DB2ENTRY.

THREADLIMIT
specifies the maximum number of threads for this DB2ENTRY that the CICS DB2 attachment allows active before requests are made to wait, are rejected, or are routed to the pool.

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, the transaction waits until one becomes available.

NOTWAIT
If all threads are busy, the 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.

All parameters on SET DB2ENTRY can be set whilst the CICS DB2 attachment is active and the transactions are active.

CEMT SET DB2TRAN

Specify this command to set the attributes of a particular DB2TRAN associated 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 matches the name of 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 DFH*xxxx*, where xxxx is the transid.

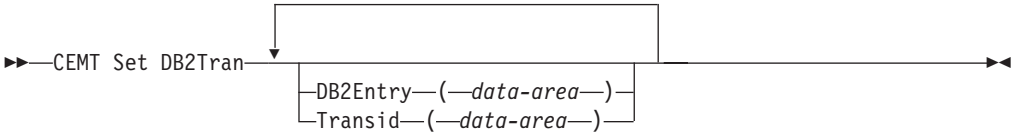
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction

- Type `CEMT SET DB2TRAN` (the minimum abbreviation is `CEMT S DB2T`) with either a value corresponding to a DB2 transaction name or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE DB2TRAN. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET DB2TRAN` (`CEMT S DB2T`) with either the name of a DB2TRAN or ALL, followed by one or more attribute settings that you wish to change. For example, `cemt s db2t all db2e(entry1)` will set all DB2TRANs to refer to a DB2ENTRY named entry1.

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET DB2TRAN



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. 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.

CEMT SET DELETSHIPPED

Change system settings that control the CICS timeout delete mechanism.

Syntax

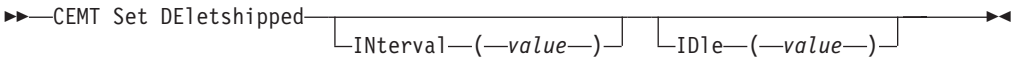
The CICS timeout delete mechanism is invoked at user-specified intervals to remove any shipped terminal definitions that have not been used for longer than a user-specified time.

SET DELETSHIPPED allows you to change the values that control the mechanism.

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET DELETSHIPPED` (the minimum abbreviation is `CEMT S DE`). You get a display that lists the current status, similar to that obtained by CEMT INQUIRE DELETSHIPPED. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET DELETSHIPPED` (`CEMT S DE`), followed by one or more attributes that you wish to change. For example, `cemt s de idl(0150000)` specifies that shipped terminal definitions are to remain installed for at least fifteen hours after they become inactive.

CEMT SET DELETSHIPPED



Options

IDle(*value*)
specifies, in the form "**0hhmmss+**", the minimum time that an *inactive* shipped terminal definition must remain installed in this region. When the CICS timeout

delete mechanism is invoked, only those shipped definitions that have been inactive for longer than this time are deleted.

The time interval can be in the range 00–99 hours; 00–59 minutes; and 00-59 seconds.

At CICS startup, the interval is set to the value specified on the DSHIPIDL system initialization parameter.

INterval(*value***)**
specifies, in the form "**0hhmmss+**", the interval between invocations of the CICS timeout delete mechanism. The timeout delete mechanism removes any shipped terminal definitions that have not been used for longer than the time specified by the IDLE option.

The time interval can be in the range 00–99 hours; 00–59 minutes; and 00–59 seconds. If you specify 0, the timeout delete mechanism is not invoked.

At CICS startup, the interval is set to the value specified on the DSHIPINT system initialization parameter. By resetting the interval, you can change the time of day at which a mass delete operation takes place.

Note: The revised interval starts from *the time the command is issued*, **not** from the time the timeout delete mechanism was last invoked, nor from the time of CICS startup.

CEMT SET DISPATCHER

Change the system values used by CICS dispatcher.

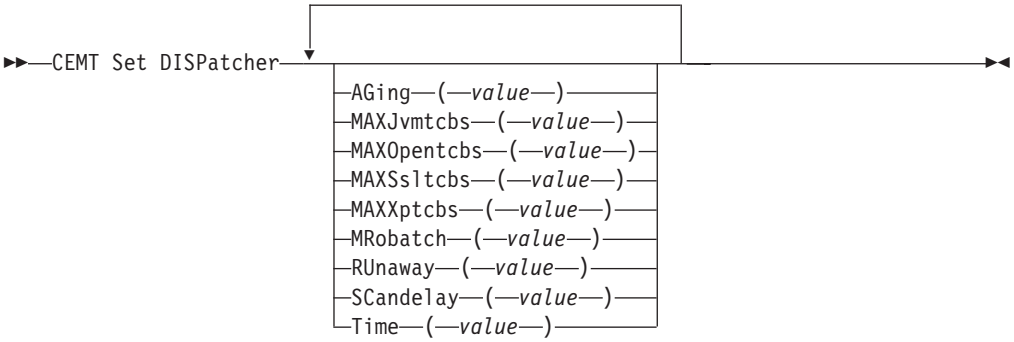
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET DISPATCHER`. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE DISPATCHER. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET DISPATCHER` followed by one or more attributes that you wish to change. For example, `cemt s disp maxo(value) maxt(value)` will reset the values for maximum number of L8 and L9 mode TCBs and the maximum number of tasks.

Typing `?` at the beginning of either the first or second line gives a syntax prompt.

CEMT SET DISPATCHER



Options

AGing (*value***)**
specifies the time factor for CICS to use in the priority aging algorithm for incrementing the priority of a task. The value is expressed as “milliseconds per unit of priority”, and must be in the range 00000–65535. The priority aging factor is defined initially by the PRTYAGE system initialization parameter, in which the value represents the number of milliseconds that must elapse before the priority of a waiting task can be adjusted upward by 1. See the *CICS System Definition Guide* for information about the PRTYAGE system initialization parameter.

MAXJvmtcbs(*value***)**
specifies 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. 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 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(*value***)**
specifies 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 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(*value***)**
specifies the maximum number of S8 mode open TCBs that can exist concurrently in the CICS region. The value specified can be in the range 1–1024.

MAXXptcbs(*value***)**
specifies 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 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 (*value***)**
specifies the number of MRO requests from connected regions that are to be batched before this region is posted. The value is in the range 001–255.

RUnaway (*value***)**
specifies the interval, in milliseconds, for which a task can have control before it is assumed to be looping.

You can specify 0, or a value in the range 500–2700000. The value chosen is rounded down to a multiple of 500. CICS purges a task if it has not given up

control after this interval (that is, if the task appears to be looping). If you specify zero, runaway task control is inoperative (that is, tasks do not get purged if they appear to be looping). The value is independent of, and can be less than, the value in the TIME operand.

CICS runaway-task detection is based upon task time, that is, the interval is decremented only when the task has control of the processor. You do not, therefore, need to allow for the effect of other jobs when setting the value.

SCandelay (*value***)**
specifies the maximum number of milliseconds between the receipt of a VTAM request, and the time when CICS begins to process the request. The number must be in the range 0–5000, and must be less than or equal to the TIME value.

In the summary report of the dispatcher statistics, SCANDELAY is referred to as ICVTSD.

Time (*value***)**
specifies the interval, in milliseconds, for which CICS releases control to the operating system if no transactions are ready to resume processing. This interval is known as the 'region exit interval'.

It can have a value in the range 100–3600000, and must be greater than or equal to the SCANDELAY value.

In the summary report of the dispatcher statistics, TIME is referred to as ICV.

CEMT SET DOCTEMPLATE

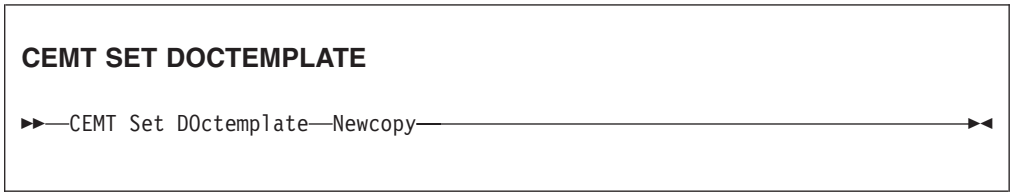
Refresh the cached copy of a document template installed in your CICS region, or phase in a new copy of a CICS program or exit program that is defined as a document template.

Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET DOCTEMPLATE` followed by one or more DOCTEMPLATE resource definition names or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE DOCTEMPLATE.
- Type `CEMT SET DOCTEMPLATE`, followed by one or more DOCTEMPLATE resource definition names or ALL, followed by NEWCOPY.

Typing `?` at the beginning of either the first or second line gives a syntax prompt.



Options

ALl
The action you request is performed for all resources of the specified type that you are authorized to access.

(*value***)**
specifies the 1–8 character name of the DOCTEMPLATE resource definition.

Newcopy
specifies that if a cached copy of the document template exists, it is to be deleted. If the document template resides in a CICS program or exit program, a new copy of the program is to be phased in. If caching is required for the document template, a new copy of the document template is to be loaded into the cache.

CEMT SET DSAS

Change the system storage attributes.

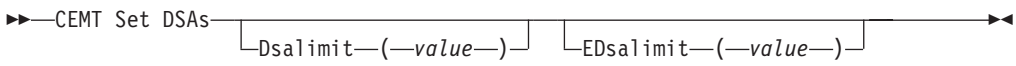
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET DSAS`. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE DSAS. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET DSAS` followed by one or more attribute settings that you wish to change. For example, `cemt s dsa ds(value) ed(value)` will reset the values for dynamic storage below the 16MB boundary and above the 16MB boundary.

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET DSAS



Options

Dsalimit(*value***)**
specifies the maximum amount of storage, as a total number of bytes, within which CICS can dynamically allocate storage for the four individual DSAs that reside below the 16MB boundary. See the *CICS System Definition Guide* for a description of the DSALIM system initialization parameter.) 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 dynamic storage is freed in the individual DSAs.

EDsalimit(*value***)**
specifies the maximum amount of storage, as a total number of bytes, within which CICS can dynamically allocate storage for the four individual DSAs that reside above the 16MB boundary. See the *CICS System Definition Guide* for a description of the EDSALIM system initialization parameter.) 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 dynamic storage is freed in the individual DSAs.

CEMT SET DSNAME

Change the status of an external data set.

Description

With the SET DSNAME command, you can change the status of a VSAM base data set. You can use the command to perform the following actions on a data set:

- 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 because of the failures of this data set (other than in-doubt failures).
- Force any UOWs that are shunted because of 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.

The SET DSNAME command does not distinguish between data sets accessed in RLS and non-RLS mode.

The new options 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, do not also specify UNAVAILABLE or QUIESCED, because this would cause backout retries to fail. UNQUIESCE should not be specified with other attributes, because the other options may fail.

Some of the attributes of a data set cannot be set until the first file that references the data set has been opened. Where an attribute is not valid until a file has been opened, an error is returned. Note that QUIESCESTATE is an attribute that can be set before any files have been opened against the specified data set.

Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

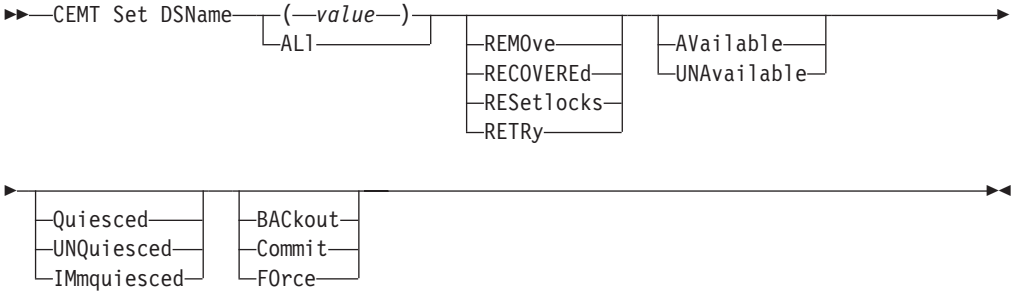
- Type `CEMT SET DSNAME` (the minimum abbreviation is `CEMT S DSN`) with either a value corresponding to a data set name or ‘ALL’. You get a display that lists the

current status, similar to that obtained by CEMT INQUIRE DSNAME. You can then tab to the highlighted or blank fields and overtype them with the required values.

- Type `CEMT SET DSNAME` (`CEMT S DSN`) with either a dataset_name or 'ALL', followed by one or more attributes that you wish to change. For example, `cemt s dsn(applb.*) q` quiesces all data sets with a high-level qualifier of `APPLB` in all CICS regions within the sysplex.

Resetting the values takes effect immediately.

CEMT SET DSNAME



Options

(*value***)**
specifies one or more names (1-44 characters) of an external data set.

ALl
If you specify the ALL operand on a SET command, any change you request is made to all resources of the specified type that you are authorized to access.

AVailable
specifies whether the data set is to be marked in this CICS region as available for use. This command sets the availability indicator, which is a local flag that a CICS region maintains in a data set name block (DSNB) for each data set. CICS can issue either RLS or non-RLS open requests for this data set.

BACkout
Specifies that all shunted in-doubt UOWs that hold locks on this data set should be backed out.

Commit
Specifies that all shunted in-doubt UOWs that hold locks on this data set should be committed.

FOrce
Specifies that all shunted in-doubt UOWs that hold locks on this data set should be forced to BACKOUT or COMMIT as specified by the ACTION attribute on the transaction resource definition.

IMmquiesced
Causes all existing CICS open RLS-mode files throughout the sysplex to be closed and the data set to be 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 RLS-mode files can open against this data set, but non-RLS-mode open requests are permitted.

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, if it was set UNENABLED by a QUIESCED or IMMQUIESCED action, but *not* if the UNENABLED state is due to 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
Causes all existing CICS files open in RLS mode throughout the sysplex to be closed and the data set to be 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 they failed in-doubt, or
- Shunted because they failed commit, or
- Shunted because they failed backout.

No files can open in RLS mode against this data set, but non-RLS open requests are permitted.

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, if it was set UNENABLED by a QUIESCED or IMMQUIESCED action, but *not* if the UNENABLED state is due to some other event. This state change is recorded in the CICS global catalog.

RECOVERED
The data set has been restored from a backup version and forward recovery has run and completed successfully. CICS normally updates the backup-while-open (BWO) attributes for the base cluster data set BASEDSNAME, using MVS/DFP 3.2 Callable Services.

If the BWO attributes are not updated after the restore of a backup copy of a data set that was taken using BWO, CICS fails a subsequent file open because the data set is marked as down-level.

REMOve
This option is used when a data set is no longer required on the local system. It deletes the data set name block for the named data set. The REMOVE option can be used only when there are no file control resource definition entries that refer to the specified data set. If any file definitions refer to the data set name when you enter SET DSNAME REMOVE, you receive an error message indicating that the use count of the data set name block is not zero. To ensure that no file definitions refer to the data set name:

- Close all files that refer to the data set. These files can be found using the command:

```
CEMT INQUIRE FILE(*) DSNAME(dsname)
```

- Either remove the data set name from each file entry (by over-typing the data set name with blanks on the CEMT display), or discard the file definition entirely.

If you have a base data set with one or more alternate indexes, you are unable to remove the data set name block for the base until you have removed all references to the base data set and all references to paths.

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.

RESETLOCKS applies only to shunted UOW log records that are classed as backout-failed and commit-failed. Therefore it cannot be issued for a BDAM data set. Backout-failed and commit-failed log records are not shunted for BDAM data sets.

You might 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.

Note:

1. This option does not apply to shunted *in-doubt* UOWs. 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 FORCEUOW.
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
Specifies that shunted UOW log records, caused by failed backout and commit processing as a result of the failure of this data set, should be retried. This is similar in operation to the EXEC CICS 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- 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.

Messages, issued at the time of a data set failure that causes UOWs to be shunted, recommend the actions that are required to recover from the failure.

RETRY does not affect data integrity, and can be used safely at any time to enable some failed recovery work to complete.

UNAvailable
Marks the data set as unavailable, and prevents the data set being opened in either RLS or non-RLS modes. 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.

UNQuiesced
Causes the data set to be marked as unquiesced in the ICF catalog. RLS or non-RLS mode files can be opened against this data set, the access mode (RLS or non-RLS) being established by the first open.

If a file has been set UNENABLED by a SET DSNAME IMMQUIESCED or QUIESCED command, UNQUIESCED sets it back to ENABLED, provided that there have been no other explicit changes to the enablement state in the meantime. This state change is recorded in the CICS global catalog.

CEMT SET DUMPDS

Change the attributes of a CICS dump data set.

Description

Using SET DUMPDS, you can open or close the active CICS dump data set, and specify whether or not the automatic switch to the inactive dump data set occurs when the active dump data set is full.

For more information about dumps, see the *CICS Problem Determination Guide*.

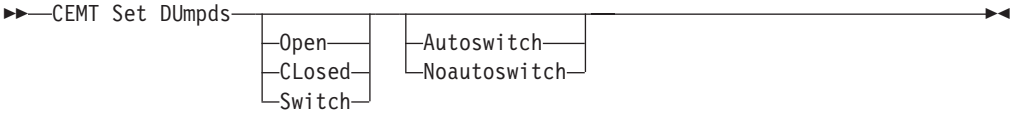
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET DUMPDS` (the minimum abbreviation is `CEMT S DU`). You get a display that lists the current status, similar to that obtained by CEMT INQUIRE DUMPDS. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET DUMPDS` (`CEMT S DU`) followed by one or more attribute settings that you wish to change. For example, `cemt s du o a` will open the active dump data set and cause autoswitch to occur only when the active dump data set is full.

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET DUMPDS



Options

Autoswitch
The switch occurs **only** when the active dump data set is full, but not subsequently.

CLosed
The active CICS dump data set is closed.

Noautoswitch
When the active dump data set is full, there is not to be an automatic switch to the inactive dump data set.

Open
The active CICS dump data set is opened.

Switch
The currently active dump data set is to become inactive, and the currently inactive data set is to become active.

CEMT SET ENQMODEL

Set the status of enqmodel definitions on the local system.

Description

Each ENQMODEL definition may be in one of the following states

ENABLED
Matching ENQ requests will be processed in the normal way.

DISABLED
Matching ENQ requests will be rejected, and the issuing task abended abcode ENQ_DISABLED. Matching INSTALL/CREATE/DISCARD requests will be processed.

WAITING
Matching ENQ requests will be rejected, and the issuing task abended abcode ENQ_DISABLED. There are INSTALL/CREATE/DISCARD requests waiting to be processed.

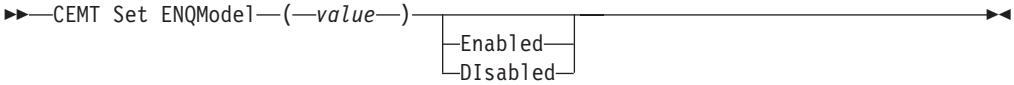
This command provides a means for you to set the ENABLED and DISABLED states for ENQmodel resources installed on the local system. An enqmodel must be enabled to allow matching EXEC ENQ requests to be processed. It must be disabled to allow a more specific enqmodel to be enabled.

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 message DFHNQ0107 is issued.

You cannot enable/disable an enqmodel which is in the waiting state.

Syntax

CEMT SET ENQMODEL



Options

(*value***)**
The 8-character identifier of the resource definition

DISABLED
The ENQMODEL will be put into the WAITING state (See Inquire Enqmodel) 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 abended abcode ENQ_DISABLED.

ENABLED
If the enqmodel is DISABLED, it will be ENABLED. Once enabled, matching ENQ requests will be processed in the normal way.

CEMT SET FEPI

Comments

For information on **CEMT SET FECONNECTION**, **CEMT SET FEPOOL**, **CEMT SET FEPROPSET**, and **CEMT SET FETARGET**, see the *CICS Front End Programming Interface User's Guide*.

CEMT SET FILE

Change some of the attributes of one or more VSAM or BDAM files.

Syntax

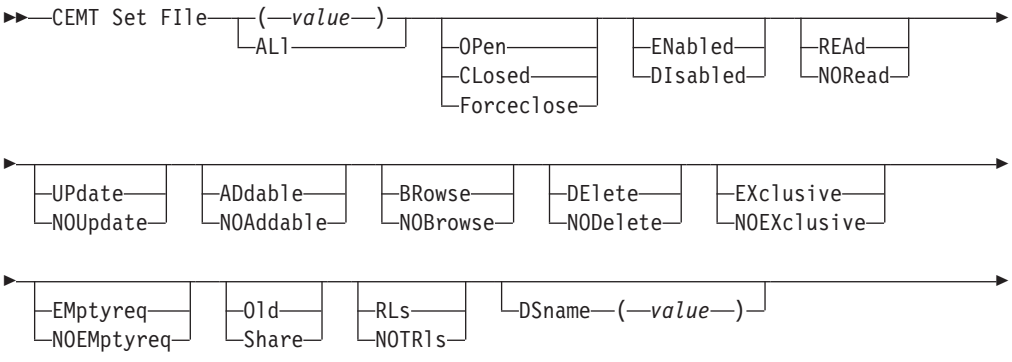
Press the Clear key to clear the screen. There are two ways of commencing this transaction:

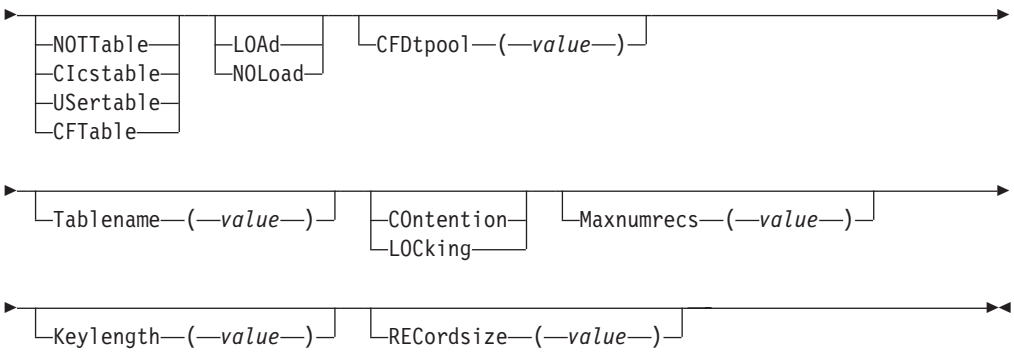
- Type `CEMT SET FILE` (the minimum abbreviation is `CEMT S FI`) with either a value corresponding to a file name or ALL. You get a display that lists the current status, similar to that obtained by **CEMT INQUIRE FILE**. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET FILE` (`CEMT S FI`) with either the name of a file or ALL, followed by one or more attribute settings that you wish to change. For example, `cemt s fi al op en` attempts to set all files open and enabled for use.

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Reset values take effect immediately.

Note: Some attributes can be changed only when the specified file is closed and disabled. In these cases, the reset values take effect when the file is enabled for use again.

CEMT SET FILE





Options

(*value***)**
One or more names (1-8 characters) defined in the file control table (FCT).

ADdable
You can add records to the file.

ALl
Any change you request is made to all files that you are authorized to access.

BRowse
You can browse records in the file.

CFDtpool (*value***)**
specifies the name of the CFDT pool in which the coupling facility data table resides.

CFtable
The file name represents a coupling facility data table.

CIcstable
The file name represents a CICS-maintained data table.

CLosed
The file is closed; the access method does not allow accesses to the data until the file is opened. The file can be opened either explicitly by the user or, if the file has the ENABLED attribute, implicitly by CICS on the next reference to the file.

When the SET FILE CLOSED command is used, the file is closed and also disabled (set UNENABLED). (A file disabled in this way prevents access requests from implicitly opening the file again.) The close is effected at the time of the command only if there are no tasks currently accessing the file. If there are current users, the display indicates CLOSE requested. This means that, when the last user finishes using the file, it effects the close.

A file that has been disabled by a SET FILE CLOSED command becomes enabled again by a subsequent SET FILE OPEN command. To distinguish a file that has been disabled by a SET FILE CLOSED from one that has been disabled by a SET FILE DISABLED command, the SET FILE CLOSED command closes the file with the UNENABLED attribute.

A file with the DISABLED attribute can be enabled only by a SET FILE ENABLED command. A file with the UNENABLED attribute can be enabled by a SET FILE ENABLED or SET FILE OPEN command.

COntention (CFDT only)
The coupling facility data table 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.

DElete
You can delete records from the file.

DIsabled
The file is not available for use by transactions except for those that are currently using it. If there are any such users, 'BEING DISABLED' is also displayed. The **CEMT SET FILE DISABLED** command has no effect on existing users; it simply prevents new users from accessing the file.

The file can be reenabled by a **SET FILE ENABLED** command. (See also UNENABLED.)

It is not possible to disable a remote file.

DSname (*value***)**
specifies the name (1–44 characters) of the VSAM or BDAM data set with which this file is associated. CEMT SET FILE(*value*) DSNAME(*value*) associates the data set with the file and causes the FILECOUNT of the data set to be incremented by 1. You can use **CEMT SET FILE DSNAME** with a null DSNAME value to dissociate a file from its current data set.

Note: DSNAME replaced OBJECTNAME.

EMptyreq (*vsam only***)**
Applies only to a VSAM file that has been defined to VSAM as a cluster with the REUSE parameter, and which is opened in non-RLS access mode. It indicates that when a file is next opened, its data is erased.

If the file is opened in RLS access mode this parameter has no effect.

A **CEMT SET FILE EMPTYREC** command can be applied only to files that are closed and either disabled or unenabled, and has no effect on the data until the file is opened. A **SET FILE EMPTYREQ** command can be canceled by a **SET FILE NOEMPTYREQ** command.

ENabled
The file is available for use by transactions and, if closed, it is opened on the first request. If the command CEMT SET FILE(*value*) CLOSED ENABLED is given while the file is in use, the status of the file becomes 'closed-unenabled'. If the command CEMT SET FILE(*value*) CLOSED is given while the file is in use, the file is disabled to prevent new users accessing the file.

EXclusive
Records on the file are protected from simultaneous update by being placed under exclusive control when a read for update is issued.

Forceclose
Similar to SET FILE CLOSED, except that any existing users of the file are abended (abend code 'AKC3' or 'ATCH'). As for CLOSED, the close is effected by the last of these users.

Note: Closing a file using the FORCECLOSE 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 FORCECLOSE option should be restricted to exceptional circumstances.

Keylength (*value*)
Specifies the key length of records in a coupling facility data table that does not require pre-loading from a source data set. 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.

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.

LOCking (CFDT only)
The coupling facility data table is to use the locking model, in which records are locked when they are updated.

Maxnumrecs (*value*)
specifies the maximum number of records that the data table for this file can hold.

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').

Note: The new value takes effect only when a new data table is created. This happens on each file open for a CICS-maintained or user-maintained data table, but only when the table does not already exist for a CF data table. Once a coupling facility data table has been created, you can change the maximum number of records only through the CFDT server MVS command interface.

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 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 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.

NOAddable
You cannot add records to the file.

NOBrowse
You cannot browse records in the file.

NODelete
You cannot delete records from the file.

NOEMptyreq (*vsam only*)
Applies only to VSAM files. It indicates that when a file is next opened, its data is not erased.

A **CEMT SET FILE NOEMPTYREC** command can be applied only to files that are closed and either disabled or unenabled, and cancels the implicit request in a previous **SET FILE EMPTYREQ** command.

NOEXclusive
Records on the file are not protected from simultaneous update.

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 entirely by the transactions that use it.

NORead
You cannot read records in the file.

NOTRls (*vsam only***)**
The file is defined to be accessed in a non-RLS mode (LSR or NSR).

For information about switching from RLS to non-RLS mode, see the *CICS Recovery and Restart Guide*.

NOTTable
The file name does not represent a data table.

NOUpdate
You cannot update records in the file.

Old
The disposition of the data set associated with the file is defined as unshared for dynamic allocation.

OPen
The file is open. If the file is OPEN ENABLED, it is available for data accesses by CICS transactions. If it is OPEN DISABLED, it must first be enabled before it is available.

If the file was unenabled by a previous SET FILE CLOSED command, the **SET FILE OPEN** command reenables the file implicitly.

REAd
You can read records in the file.

RECordsize
specifies the maximum record size for a coupling facility data table that does not require pre-loading from a source data set.

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 1 to 32767 bytes, or specify a record size of zero to remove a previously defined value.

RLs (*vsam only***)**
The file is defined to be accessed in RLS mode.

Share
The disposition of the data set associated with the file is defined as shared for dynamic allocation.

Tablename (*value***)**
specifies the name of the table for a coupling facility data table.

UPdate
You can update records in the file.

USertable
The file name represents a user-maintained data table.

CEMT SET GTFTRACE

Change the status of GTF tracing

Description

SET GTFTRACE allows you to switch CICS tracing to the MVS Generalized Tracing Facility (GTF) on and off. This command is intended for use at terminals that do not support the CETR full-screen tracing transaction.

For guidance on using GTF tracing, see the *CICS Problem Determination Guide*.

Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET GTFTRACE` (the minimum abbreviation is `CEMT S G`). You get a display that lists the current status, similar to that obtained by CEMT INQUIRE GTFTRACE. You can then tab to the highlighted field and overtype it with the required value.
- Type `CEMT SET GTFTRACE` (`CEMT S G`) followed by the attribute setting that you wish to change. For example, `cemt s g sta` will cause CICS GTF tracing to be started.

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET GTFTRACE

Options

STArt
CICS tracing to the MVS Generalized Tracing Facility (GTF) is started.

STOp
CICS tracing to the MVS Generalized Tracing Facility (GTF) is stopped.

CEMT SET HOST

Enable or disable a virtual host.

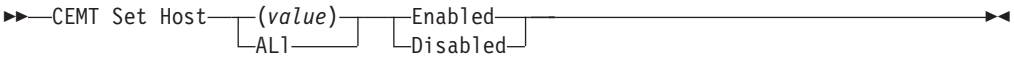
Description

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. When a virtual host is disabled, CICS returns an HTTP response with a 503 (Service Unavailable) status code to Web clients.

When you inquire on an individual URIMAP definition, a special status HDISABLED 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 HDISABLED status cannot be discarded. If you want to discard the definition, it must be disabled individually (using the SET URIMAP command).

Syntax

CEMT SET HOST



Options

(*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 same name. A host name in a URIMAP definition can be up to 120 characters.

Enabled
The URIMAP definitions that make up the virtual host can be accessed.

Disabled
The URIMAP definitions that make up the virtual host cannot be accessed.

CEMT SET INTTRACE

Switch CICS internal tracing on or off.

Description

SET INTTRACE allows you to switch CICS internal tracing on and off. This command is intended for use at terminals that do not support the full-screen tracing transaction CETR.

Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET INTTRACE` (the minimum abbreviation is `CEMT S IN`). You get a display that lists the current status, similar to that obtained by CEMT INQUIRE INTTRACE. You can then tab to the highlighted field and overtype it with the required value.
- Type `CEMT SET INTTRACE` (`CEMT S IN`) followed by the attribute setting that you wish to change. For example, `cemt s in sta` will cause CICS internal tracing to be started.

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET INTTRACE

Options

STArt
CICS internal tracing is started.

STOp
CICS internal tracing is stopped.

CEMT SET IPCONN

Change the attributes of an IPCONN or cancel outstanding AIDs.

Description

SET IPCONN allows you to change some of the attributes of an IPCONN.

Note: See also “CEMT SET CONNECTION” on page 375. The CEMT SET CONNECTION command is used to change the attributes of MRO and ISC over SNA connections.

For information about the different kinds of intercommunication connections, see the *CICS Intercommunication Guide*.

If you want to install a new IPCONN definition when one is already installed, you must set the connection OUTSERVICE before using the CEDA INSTALL commands for your new definition. See the *CICS Resource Definition Guide* for further information about connections.

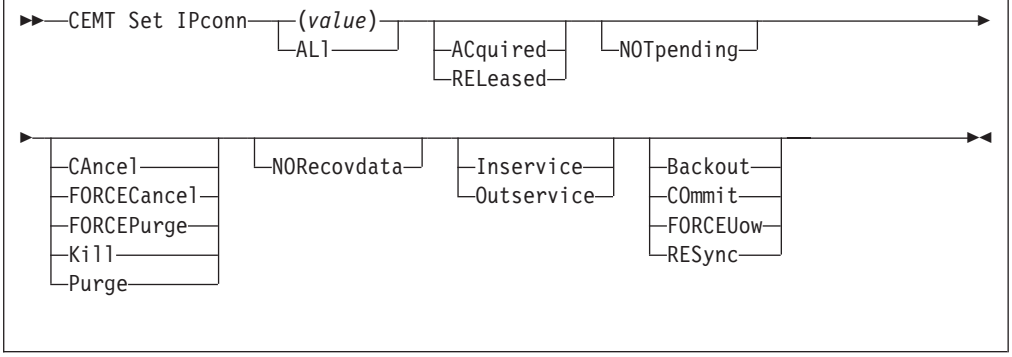
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET IPCONN` (the minimum abbreviation is `CEMT S IP`) followed by one or more connection identifiers or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE IPCONN. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET IPCONN` (`CEMT S IP`) followed by one or more connection identifiers or ALL, followed in turn by one or more attribute settings that you wish to change. For example, `cemt s ip al i` resets the values for all connections to make them available for use (inservice).

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET IPCONN



Options

(*value***)**
specifies the name (1–8 characters) of the connection to be modified. This is the name of the remote system or region specified on the IPCONN option of the IPCONN definition. You can specify more than one name.

ACquired
specifies that CICS is to acquire a session with the system represented by the IPCONN name. A connection cannot be both ACQUIRED and OUTSERVICE.

ALl
specifies that any changes you request are to be made to all resources of the specified type that you are authorized to access.

Backout
specifies that all UOWs shunted due to the failure of this connection are to be backed out. The normal resynchronization process is partially overridden: decisions are taken for any units of work that are in-doubt due to a failure of the connection; but the decisions are recorded and any data inconsistencies are reported when the connection is next acquired.

CAncel

specifies that queued requests by transactions to use this IPCONN are to be canceled.

Queued requests to use this IPCONN by CICS system transactions that manage communications across the IPCONN are not purged unless FORCECANCEL is specified.

Message DFHISnnnn is written to CSMT to indicate how many queued requests to use this IPCONN have been deleted for the IPCONN and how many remain.

A “QUEUED REQUESTS CANCELED” message appears on the CEMT panel whenever queued requests to use this IPCONN are deleted using the CANCEL option of the CEMT SET IPCONN command.

COmmit
specifies that all UOWs shunted due to the failure of this connection are to be committed. The normal resynchronization process is partially overridden: decisions are taken for any units of work that are in-doubt due to a failure of the connection; but the decisions are recorded and any data inconsistencies are reported when the connection is next acquired.

FORCECancel
specifies that all queued requests by transactions to use this IPCONN are to be canceled, including requests by CICS system transactions that manage communications across this IPCONN. This can lead to unpredictable results and should be used only in exceptional circumstances.

A “QUEUED REQUESTS CANCELED” message appears on the CEMT panel whenever queued requests to use this IPCONN are deleted using the FORCECANCEL option of the CEMT SET IPCONN command.

FORCEPurge
specifies that all transactions running on sessions to the connected system are to be abnormally terminated immediately. This can lead to unpredictable results and should be used only in exceptional circumstances.

In some extreme cases (for example, if an error occurs during backout processing), CICS might terminate abnormally.

FORCEUow
specifies that all UOWs shunted due to the failure of this connection are to be forced to back out or commit, as specified in the ACTION option of the TRANSACTION definition. The normal resynchronization process is partially overridden: decisions are taken for any units of work that are in-doubt due to a failure of the connection; but the decisions are recorded and any data inconsistencies are reported when the connection is next acquired.

Inservice
specifies that the system is to be placed in service; that is, available for use.

Kill
specifies that the task is to be terminated. System and data integrity is not guaranteed. The KILL option extends the PURGE and FORCEPURGE options. You should use it only after an attempt has been made to PURGE or FORCEPURGE a task. The KILL option does not guarantee integrity of any kind but in some situations it allows you to free up a stalled region, enabling the region to continue processing. In some cases, for example if a task is killed during backout processing, CICS terminates abnormally.

NORecovdata
specifies that the normal resynchronization process is to be overridden. NORECOVDATA forces in-doubt units of work (according to the transaction definitions), targets any resynchronization tasks that are outstanding for the connection, and erases the logname previously received from the partner system. The state of the connection is reset.

Note: You should use SET IPCONN NORECOVDATA only in exceptional circumstances. It erases recovery information and may compromise data integrity for units of work that have updated resources on remote systems.

Examples of circumstances in which you might need to use it are:

- You need to discard a connection and it is not possible for the quiesce protocols with the partner system to be completed.
- An operational or logic error results in a logname mismatch for the connection. The connection state must be reset to allow the exchange lognames process to complete.

Note: If you specify NORECOVDATA you cannot specify COMMIT, BACKOUT, FORCEUOW, RESYNC, or NOTPENDING.

NOTpending
For a connection to a CICS Transaction Server for z/OS partner that has performed an initial start, specifies that the normal resynchronization process is to be overridden.

NOTPENDING forces all in-doubt units of work (according to the transaction definition) that were created by the connection before the initial start of the partner. It also forgets any resynchronization tasks (waitforget UOW-links) that are outstanding for the connection, and created before the initial start of the partner.

The PENDING condition indicates the existence of recovery information (either shunted UOWs or decisions remembered for the partner) on a connection that has experienced a lognames mismatch with its partner. This indicates that the partner has performed an initial start and that the recovery protocol has been corrupted by a loss of log data at the partner.

It is not possible to set a connection to NOTPENDING state (forcing in-doubt and erasing NOFORGET UOWs) until CICS has made contact with the partner and received a new logname from it.

Decisions for a whole connection can be forgotten, but that does not affect the memory of a decision for any other connection involved in the UOW.

Note: If you specify NOTPENDING you cannot specify COMMIT, BACKOUT, FORCEUOW, RESYNC, or NORECOVDATA. For advice on which option to use, see the notes following the description of the RESYNC option.

The exchange lognames function and the resynchronization function are described in the *CICS Intercommunication Guide* and the *Systems Network Architecture—LU6.2 Reference: Peer Protocols* manual.

Outservice
Place the system out of service; that is, unavailable for use.

Purge
specifies that transactions running on the connected system are to be abnormally terminated. Transactions are terminated only if system and data integrity can be maintained. A transaction is not purged if its definition specifies SPURGE=NO.

RELeased
specifies that CICS is to release a session with the system represented by the IPCONN name.

RESync
specifies that any UOWs shunted due to 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. The normal resynchronization process is partially overridden: decisions are taken for any units of work that are in-doubt due to a failure of the connection; but the decisions are recorded and any data inconsistencies are reported when the connection is next acquired.

Note:

1. The COMMIT, BACKOUT, FORCEUOW and RESYNC operations are synchronous with setting the state of the UOW; that is, an INQUIRE UOW following SET IPCONN BACKOUT, COMMIT, FORCEUOW, or RESYNC returns the new UOW states.

2. Specifying one of these options unshunts all units of work that have failed due to a failure of the connection. Before issuing SET IPCONN FORCEUOW, 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 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.
3. You can specify only one of the BACKOUT, COMMIT, FORCEUOW, RESYNC, NOTPENDING, and NORECOVDATA options. SET IPCONN NORECOVDATA should be used only in exceptional circumstances.
4. To force all in-doubt units of work caused by a failure of the connection in the same direction, use SET IPCONN COMMIT or SET IPCONN BACKOUT.
5. The BACKOUT, COMMIT, FORCEUOW, or RESYNC options of SET IPCONN and SET UOW do not clear resync information. If you want to do this, you must use SET IPCONN NOTPENDING or SET IPCONN NORECOVDATA.
6. You can issue BACKOUT, COMMIT, FORCEUOW, or RESYNC commands *before* issuing SET IPCONN NOTPENDING or SET IPCONN NORECOVDATA.

CEMT SET IRC

Open or close the interregion communication (IRC) facility.

Description

New MRO connections can be installed without first closing IRC. However, if you want to modify existing MRO connections, IRC must be closed using the SET IRC CLOSED command. For example:

```
CEMT SET IRC CLOSED
CEDA INSTALL GROUP(groupname)
CEMT SET IRC OPEN
```

Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET IRC` (the minimum abbreviation is `CEMT S IR`). You get a display that lists the current status, similar to that obtained by CEMT INQUIRE IRC. You can then tab to the highlighted field and overtype it with the required value.
- Type `CEMT SET IRC` (`CEMT S IR`) followed by the attribute setting that you wish to change. For example, `cemt s ir o` will cause interregion communication to be initialized.

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET IRC

Options

Closed
Interregion communication for this CICS system is to be terminated after all tasks using IRC sessions have terminated normally.

Immclose
All tasks using IRC sessions are to be abended when they next try to use the session. Interregion communication is terminated when all tasks that were using IRC have been abended or have terminated normally.

Open
Interregion communication for this CICS system is to be initialized. This allows connections to be established between this system and other systems, including DL/I batch regions and non-CICS client programs using the external CICS interface (EXCI). You can control the availability of individual connections with the INSERVICE|OUTSERVICE settings on the connection names.

CEMT SET JOURNALNAME

Enable or disable a CICS user journal.

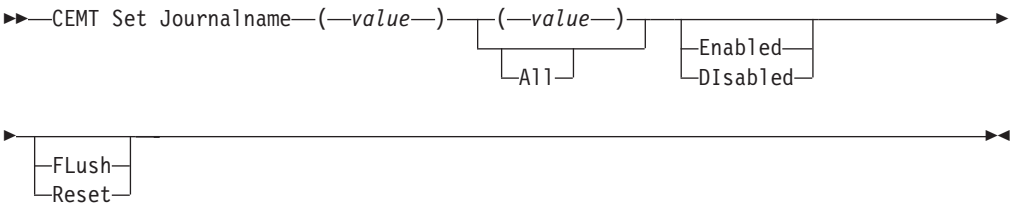
Description

The SET JOURNALNAME command allows you to enable or disable a CICS user journal. It has no effect on a journal that is being used as the forward recovery log or auto-journal for a VSAM file until the next time the file is opened or 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 (see the *CICS Resource Definition Guide*.)

Syntax

CEMT SET JOURNALNAME



Options

(*value***)**
Specifies the journal name.

To modify journals defined with a numeric identifier in the range 1–99, specify journal name DFHJ*nn*, where *nn* is the journal number.

You cannot specify DFHLOG or DFHSHUNT, because you are not allowed to modify the status of the system log.

All
specifies that any changes you request are made to all resources of the specified type that you are authorized to access.

DIsabled
The journal is disabled. It cannot be used again until it is reenabled by a SET JOURNALNAME(...) STATUS(ENABLED) or ACTION(RESET) command.

Enabled
The journal is open and is available for use.

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 the stream is processed using a batch utility.

Reset
The journal is disabled, but can be reenabled by the next SET JOURNALNAME STATUS(ENABLED) or ACTION(RESET) command for this journal.

CEMT SET JVMPOOL

Enable or disable the JVM pool.

Description

The SET JVMPOOL command allows you to enable or disable the JVM pool.

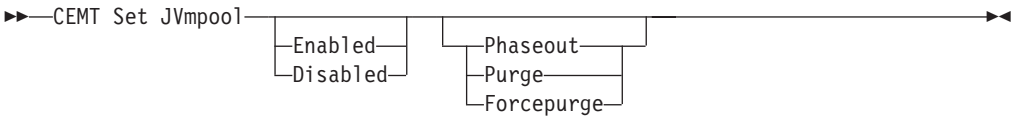
The Phaseout, Purge and Forcepurge functions of this command are now deprecated, and you should use the corresponding functions of the PERFORM JVMPOOL command instead.

Syntax

Press the Clear key to clear the screen and enter `CEMT SET JVMPOOL` (the minimum abbreviation is `CEMT S JVMP`). This gives you a display that lists the current status, similar to that obtained by CEMT INQUIRE JVMPOOL. You can then tab to the highlighted or blank fields and overtype them with the required values.

Typing `?` at the end of the first line or the beginning of the second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET JVMPOOL



Options

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.

Enabled
The pool status is set to enabled for use and Java programs can execute using JVMs from the pool.

Forcepurge
All tasks using JVMs are terminated by the SET TASK FORCEPURGE mechanism, and the JVMs are terminated. If the shared class cache is at the Java 1.4.2 level, it is deleted when all the JVMs that were using it have been terminated. This function is deprecated; use the CEMT PERFORM JVMPOOL command instead.

Phaseout
All JVMs in the pool are marked for deletion. The JVMs are actually deleted when they finish running their current Java program. If the shared class cache is at the Java 1.4.2 level, it is deleted when all the JVMs that were using it have been terminated. This function is deprecated; use the CEMT PERFORM JVMPOOL command instead.

Purge
All tasks using JVMs in the pool are terminated by the SET TASK PURGE mechanism, and the JVMs are terminated. If the shared class cache is at the Java 1.4.2 level, it is deleted when all the JVMs that were using it have been terminated. This function is deprecated; use the CEMT PERFORM JVMPOOL command instead.

CEMT SET LIBRARY

Change the attributes of LIBRARY resource.

Description

SET LIBRARY allows the ranking, critical status and enablement status of a LIBRARY to be changed.

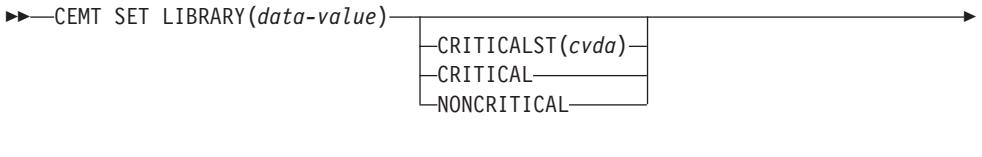
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET LIBRARY` (the minimum abbreviation is `CEMT S LIB`) followed by the LIBRARY name. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE LIBRARY. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET LIBRARY` (`CEMT S LIB`) followed by the LIBRARY name, followed in turn by one or more attribute settings that you wish to change.

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET LIBRARY



Options

CRITICALST(*cvda***)**
specifies whether the LIBRARY is critical to the start up of CICS. CVDA values are:

CRITICAL
The LIBRARY is critical to CICS startup. If the LIBRARY cannot be successfully installed during CICS startup for any reason, then a 'GO or CANCEL' message will be issued to allow the operator to decide whether to override the criticality and allow CICS to start. If the operator selects 'GO', the LIBRARY will be installed in a 'disabled' status, unless install is not possible at all; for example, due to a short-on-storage condition.

Note: If the reply is to continue with the startup, the LIBRARY will not be recatalogued as NONCRITICAL, so the critical status should be explicitly set to NONCRITICAL if it is decided that the LIBRARY should not be regarded as CRITICAL in future.

NONCRITICAL
The LIBRARY is not critical to CICS startup. If the LIBRARY cannot be successfully installed during CICS startup, then the LIBRARY will be left in an installed but disabled state and a warning message will be issued, but CICS startup will continue.

LIBRARY(*data-value***)**
specifies the 8-character name of the LIBRARY whose attributes are being changed

ENABLESTATUS(*cvda***)**
specifies whether the LIBRARY is included in the overall LIBRARY search order. CVDA values are:

DISABLED
The LIBRARY is not included in the LIBRARY search order. The data sets in this LIBRARY concatenation will not be searched for program artifacts to load.

ENABLED
The LIBRARY is included in the LIBRARY search order. The data sets in this LIBRARY concatenation will be searched for program artifacts to load.

Note: When a LIBRARY is disabled the information about where a program was loaded from becomes invalid, so when the LIBRARY is re-enabled a NEWCOPY or PHASEIN has to be issued before the program can be loaded again.

RANKING(*data-value***)**
A number between 1 and 99 which specifies where this LIBRARY should appear in the overall LIBRARY search order relative to other LIBRARY concatenations. A lower number indicates that this LIBRARY will be searched for programs to load before other LIBRARY resources with higher ranking numbers. DFHRPL has a special ranking value of 10, and this ranking value cannot be specified for a dynamic LIBRARY.

CEMT SET LINE

Change the status of a line.

Description

The SET LINE command allows you to purge or forcepurge a line, to put the line in or out of service, and to adjust the relative line number.

You can specify a list of identifiers (see "List of resource identifiers" on page 128), but you cannot use the symbols * and + to specify a family of identifiers.

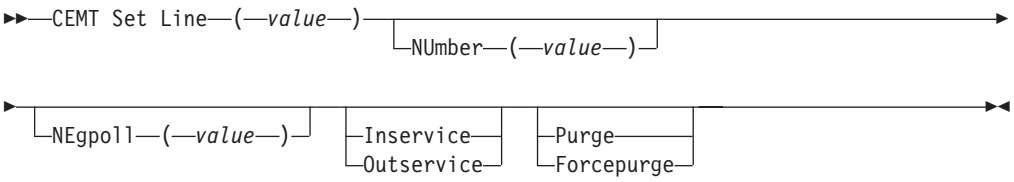
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET LINE` (the minimum abbreviation is `CEMT S L`) with a terminal identifier. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE LINE. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET LINE` (`CEMT S L`) with a terminal identifier, followed by one or more attribute settings that you wish to change. For example, `cemt s l(`*value*`) i` will make the line available for use (inservice).

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET LINE



Options

(*value***)**
specifies the identifier of a terminal on the line. This request is valid only for non-VTAM terminals.

Forcepurge
All transactions running with terminals on the line are immediately terminated

abnormally. Data integrity is not guaranteed. In some extreme cases (for example, if an error occurs during backout processing), CICS might terminate abnormally.

Inservice
The line is available for use. Changing the status of the line does not affect the equivalent status of the terminals on it.

NUmber (*value*)
specifies the relative line number of a pooled line. It is mandatory when inquiring about pooled lines. For information about pooled lines, see the *CICS Resource Definition Guide*.

Outservice
The line is not available for use. Changing the status of the line does not affect the equivalent status of the terminals on it.

Purge
Tasks running on terminals associated with the line are abnormally terminated, but task termination occurs only if system and data integrity can be maintained.

CEMT SET MODENAME

Change the status of a mode name.

Description

The SET MODENAME command is used to reset the connection to a remote system or to another CICS region, change the number of sessions available, and acquire or close the sessions associated with the mode name.

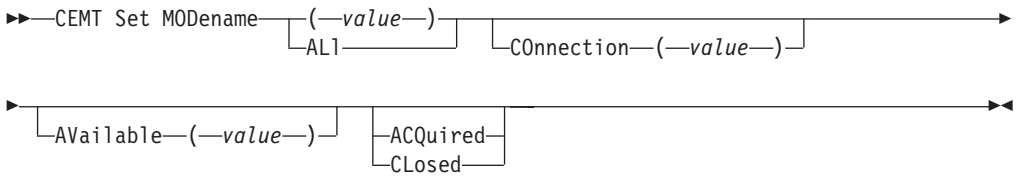
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET MODENAME` (the minimum abbreviation is `CEMT S MOD`) with either a modename or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE MODENAME. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET MODENAME` (`CEMT S MOD`) with either a modename or ALL, followed by one or more attribute settings that you wish to change. For example, `cemt s mod(modename) acq` will cause CICS to acquire all the negotiated contention-winner sessions within the group named by modename.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET MODENAME



Options

(*value***)**
specifies one or more names (1-8 characters) of a group of sessions. Mode names need not be unique. You cannot specify a list of names, but you can use the symbols * and + to specify a family of mode names.

A mode name consisting of eight blanks is valid. To specify this name, enclose the string of blanks within single quotation marks:

```
MODENAME('   ')
```

ACQuired
CICS is to acquire (or “bind”) all the negotiated contention-winner sessions within the group.

ALl
specifies that any changes you request are done to all resources of the specified type that you are authorized to access.

AVailable (*value***)**
specifies the maximum number of sessions within the group that can be concurrently allocated for use. This value has to be the same for both sides involved in the session.

CLosed
CICS is to release all sessions within the group. The sessions are quiesced and unbound. No further sessions can be acquired by another CICS system. No further activity is possible on the sessions within the group until the group is reacquired by the same CICS system in which it has been closed.

COnnection (*value***)**
specifies an identifier (1–4 characters) defined for an IRC or ISC connection.

CEMT SET MONITOR

Change the status of monitoring.

Description

SET MONITOR allows you to select the classes of monitoring data you want collected, switch monitoring on or off, and set the status of monitoring options. For programming information on the monitoring classes, see the *CICS Customization Guide*.

Note:

1. When you change the status of CICS monitoring from OFF to ON, CICS starts to accumulate monitoring data, and to write monitoring records for the active monitoring classes, for all tasks that start **after** the change is made. The monitoring status of tasks being processed is not affected.
2. When you change the status of CICS monitoring from ON to OFF, CICS stops writing monitoring data immediately. No monitoring data is accumulated for tasks that start after the change is made.
3. The monitoring class settings can be changed irrespective of whether monitoring is ON or OFF.

   If monitoring is OFF, you can change the class settings as required in preparation for the next time STATUS is set to ON.

   If monitoring is ON, changes to the class settings have the following effects:

a. Changing a class from not-active to active has no effect on the data written for running tasks.

b. Changing a class from active to not-active prevents monitoring data for that class being written for running tasks as follows:

Performance
Any performance class data remaining in the output buffer is recorded, and no further data is collected.

Exception
Monitoring data is not written at the completion of the exception condition.

Resource
Any transaction resource monitoring data remaining in the output buffer is recorded, and no further data is collected.

c. Exception class, transaction resource class, and performance class data are gathered during the life of a running transaction. When you change these settings, you should be aware of their impact on transactions that are running. For example, if a transaction is started when the performance class is set to PERF and monitoring is ON, performance class data is accumulated for that transaction. If you switch the performance class to NOPERF while the transaction is running, and then back to PERF before the transaction ends, performance data will be missing from the final monitoring data for that transaction, even though, when it started and when it ended, the performance class was set to PERF. Transaction resource monitoring is treated the same as the performance class.

4. If you change the setting for the data compression option, the new setting applies to all monitoring SMF records written from that point on, even if they are for a task being processed at the time the change is made. The new setting also applies to any records which are in the buffer waiting to be written to SMF at the time the change is made. The change only applies until a CICS restart, at which time the setting for the data compression option reverts to that specified in the monitoring control table (MCT) controlled by the DFHMCT macro.

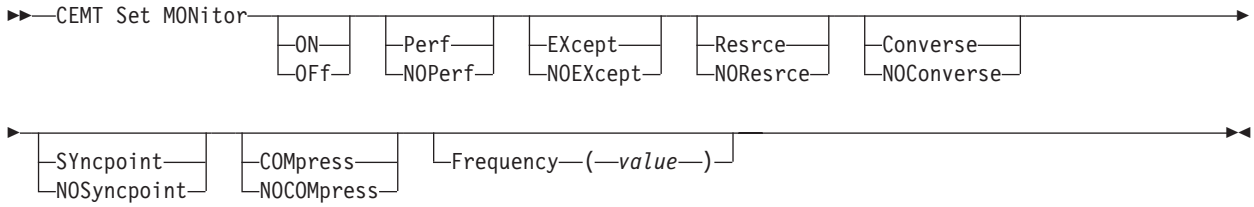
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET MONITOR` (the minimum abbreviation is `CEMT S MON`). You get a display that lists the current status, similar to that obtained by CEMT INQUIRE MONITOR. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET MONITOR` (`CEMT S MON`) followed by one or more attribute settings that you wish to change. For example, `cemt s mon on p ex` will set CICS monitoring ON and collect performance-class and exception-class data.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET MONITOR



Options

COMpress
Data compression is to be performed for monitoring records.

Converse
Conversational tasks are to have separate performance class records produced for each pair of terminal control I/O requests (Converse or Send/Receive pair).

EXcept
The exception class of monitoring data is to be collected.

Frequency (*value***)**
Specifies the interval for which monitoring produces automatically a transaction performance class record for any long-running transaction that has been in the system for a greater time than the interval. Frequency times are 0, or in the range 000100–024000. The default frequency value is 0, which means that FREQUENCY monitoring is inactive; that is, no performance class records are produced for long-running transactions.

CICS can produce a performance class monitoring record in this way only when the long-running transaction is running on the QR or CO TCBs.

NOCOMpress
Data compression is not to be performed for monitoring records.

NOConverse
Conversational tasks are not to have separate performance class records produced for each pair of terminal control I/O requests (Converse or Send/Receive pair).

NOEXcept
The exception class of monitoring data is not to be collected.

NOPerf
The performance class of monitoring data is not to be collected.

NOResrce
Transaction resource monitoring data is not to be collected.

NOSyncpoint
A transaction performance class record is not to be produced when a transaction takes an explicit or implicit syncpoint (unit-of-work), except when the syncpoint is part of task termination or a syncpoint rollback.

OFf
CICS monitoring is to be switched off. When monitoring is OFF, no monitoring data is accumulated or collected, irrespective of the settings of the monitoring data classes.

ON
CICS monitoring is to be switched on. When monitoring is ON, monitoring data is accumulated continuously, and is collected (written out to an SMF data set) for each of the classes of monitoring data that are active.

Perf
The performance class of monitoring data is to be collected.

Resrce
Transaction resource monitoring data is to be collected.

SYncpoint
A transaction performance class record is to be produced when a transaction takes an explicit or implicit syncpoint (unit-of-work), except when the syncpoint is part of task termination or a syncpoint rollback.

CEMT SET NETNAME

Change the status of networks.

Description

If you are familiar with network names, you may prefer to use this command as an alternative to CEMT SET TERMINAL (described under “CEMT SET TERMINAL” on page 451). However, CEMT SET NETNAME cannot be used for APPC sessions.

“netname” is the name by which the remote system or terminal is known to the network. The remaining operands for NETNAME are identical to those for TERMINAL.

Netnames need not be unique, and you cannot specify a list of names.

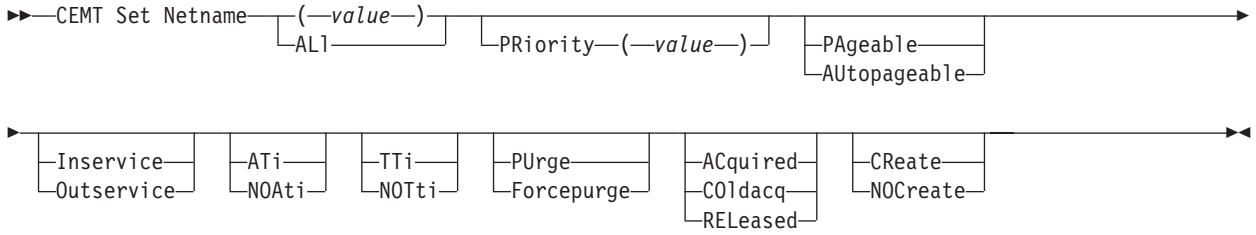
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET NETNAME` (the minimum abbreviation is `CEMT S N`) with netname or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE NETNAME. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET NETNAME` (`CEMT S AUTO`) with netname or ALL, followed by one or more attribute settings that you wish to change. For example, `cemt s n(netname) i at` will reset the named remote system or terminal to inservice and available for use.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET NETNAME



Options

(*value***)**
specifies the name by which the remote terminal is known in the network.

ACquired (*vtam only***)**
CICS is in session with the logical unit represented by the terminal.

ALl
specifies that any changes you request are made to all resources of the specified type that you are authorized to access.

ATi (*vtam only***)**
The terminal is available for use by transactions that are initiated automatically from within CICS or, if the terminal is an ISC session, by transactions that are using this session as an alternative facility to communicate with another system.

AUtopageable (*vtam only***)**
Pages after the first in a series are to be written to the terminal automatically.

COldacq (*vtam only***)**
This is a special form of ACQUIRED, where no resynchronization is required. If the previous session abended, the use of COLDACQ overrides CICS integrity control. This could lead to integrity problems. Also, you should check the CSMT log for an activity keypoint after the restart of a session following a CICS failure. If there is no activity keypoint, you should issue COLDACQ again after the next emergency restart.

CReate (*vtam only***)**
If the terminal is not in session, CICS is to acquire it if it is needed to satisfy an ATI request. This cannot be specified for IRC sessions.

Forcepurge
All transactions running with this terminal are immediately terminated abnormally. Data integrity is not guaranteed. In some extreme cases (for example, if an error occurs during backout processing), CICS might terminate abnormally.

Inservice (*vtam only***)**
The terminal is available for use. For VTAM, INSERVICE means that the terminal can be ACQUIRED.

NOAti (*vtam only***)**
The terminal is not available for use by transactions that are initiated automatically from within CICS or, if the terminal is an ISC session, by transactions that are using this session as an alternative facility to communicate with another system.

Note: A terminal cannot be defined with both NOATI and NOTTI.

NOCreate (*vtam only***)**
If the terminal is not in session, CICS is not to acquire it to satisfy an ATI request. A session must be started by, for example, a logon request or a CEMT SET TERMINAL ACQUIRED command before the ATI request can be satisfied.

If NOCREATE is set for an LU6.1 ISC session and there are no allocatable sessions left, the connection is placed OUTSERVICE.

NOTti (*vtam only***)**
This terminal cannot be used by transactions.

Note: A terminal cannot be defined with both NOATI and NOTTI.

Outservice (*vtam only***)**
The terminal is not available for use. Setting a terminal OUTSERVICE means that the terminal can no longer be used by transactions. If PURGE or FORCEPURGE is also specified, any transaction using the terminal is terminated abnormally. If PURGE or FORCEPURGE is not specified, the transaction is allowed to terminate normally, but no further transactions are allowed to use the terminal. For VTAM, setting a terminal OUTSERVICE also causes it to be released and the operator to be signed off, either immediately or when the current transaction has terminated.

In an LU6.1 ISC session the connection is set OUTSERVICE if there are no allocatable sessions left.

To set an IRC session OUTSERVICE, you must set the connection to the MRO partner OUTSERVICE.

PAgeable (*vtam only***)**
Pages after the first in a series are to be written to the terminal only if you request them to be.

PRiority (*value) (vtam only***) (VTAM only)**
specifies the priority of a terminal relative to other terminals. The priority of a task is the sum of the transaction priority, the terminal priority, and the operator priority. Priority has no meaning for terminals that are ISC sessions being used as alternative facilities. The value must be in the range 0–255, where 255 is the highest priority.

PUrge
Tasks running on this terminal are abnormally terminated, but task termination occurs only if system integrity can be maintained.

RELeased (*vtam only***)**
CICS is not in session with the logical unit represented by the terminal. Setting a terminal RELEASED causes the session to be terminated. Running transactions are allowed to finish unless PURGE or FORCEPURGE is also specified.

TTi (*vtam only***)**
This terminal can be used by transactions.

CEMT SET PIPELINE

Enable or disable a PIPELINE.

Description

If the optional attributes are omitted, the command has no effect.

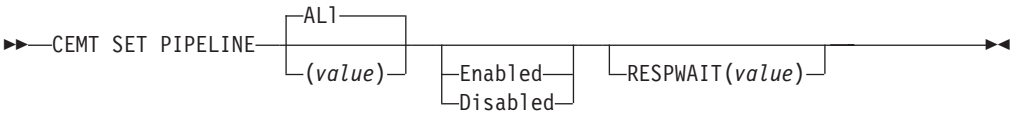
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET PIPELINE` (the minimum abbreviation is `CEMT S PI`) followed by one or more PIPELINE identifiers or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE PIPELINE. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET PIPELINE` (`CEMT S PI`) followed by one or more PIPELINE identifiers or ALL, followed in turn by one or more attribute settings that you wish to change. For example, `cemt s pi al e` resets the values for all PIPELINEs to make them available for use (enabled).

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

SET PIPELINE



Conditions: NOTAUTH, NOTFND

Options

ALl
Any changes you request are made to all resources of the specified type that you are authorized to access.

(*value***)**
specifies the 1–8 character name of the PIPELINE.

Enablestatus
Sets 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.

Respwait(*value*)
Specifies the time in seconds that an application program should wait for a response message from a remote Web service. The value can range from 0 to 9999 seconds.

If you do not specify a value, the default timeout value of the transport protocol is used.

- The default timeout value for HTTP is 10 seconds.
- The default timeout value for MQ is 60 seconds.

CEMT SET PROCESSTYPE

Change the attributes of a CICS business transaction services process-type.

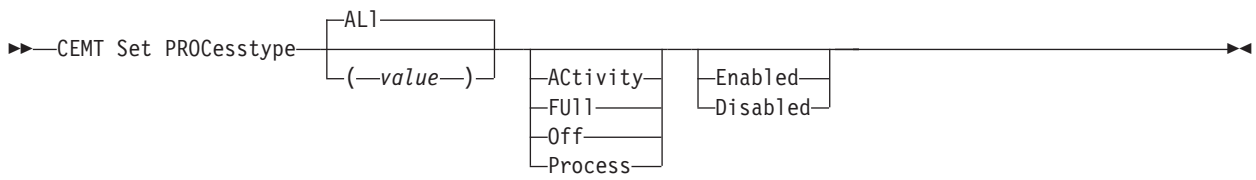
Description

SET PROCESSTYPE enables you to change the current state of audit logging and the enablement status of BTS 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 freely use the SET PROCESSTYPE command to modify your process-types. However, if you are using BTS in a CICSPlex, 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 CICSPlex.

Syntax

CEMT SET PROCESSTYPE



Options

ACtivity|FUll|Off|Process
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 values are:

ACtivity
Activity-level auditing. Audit records will be written from:

1. The process audit points
2. The activity primary audit points.

FUll Full auditing. Audit records will be written from:

1. The process audit points
2. The activity primary *and* secondary audit points.

Off No audit trail records will be written.

Process
Process-level auditing. Audit records will be 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.

ALl
specifies that any changes you specify are made to all process-types that you are authorized to access.

Enabled|Disabled
specifies whether new processes of this type can be created. The 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.

PROCesstype(value)
specifies the 8-character name of the process-type whose attributes are to be changed.

CEMT SET PROGRAM

Change the attributes of installed programs.

Description

Only programs that have been defined in the CICS system definition (CSD) file and installed on the running CICS system are accessible through CEMT.

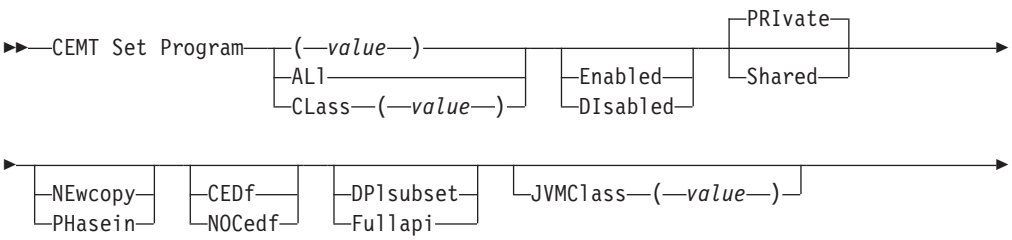
Syntax

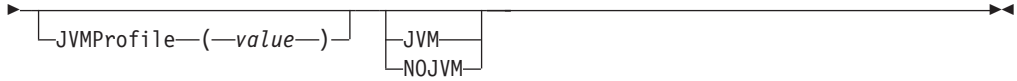
Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET PROGRAM` (the minimum abbreviation is `CEMT S PROG`) with pgrmid, CLASS(*value*), or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE PROGRAM. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET PROGRAM` (`CEMT S PROG`) followed by one or more attribute settings that you wish to change. For example, `cemt s prog(pgrmid) e ne` will reset the values for the named program to make it available for use (enabled) and a new copy of the program will be used when all the transactions currently using the program have finished (newcopy).

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET PROGRAM





Options

(*value***)**
specifies a program identifier (1–8 characters), which is the name of a specific program entry in the table of installed program definitions. If you omit the program name, the ALL option is assumed by default.

You cannot specify the CLASS option and a program name.

ALl
specifies that any changes you request are to be made to all resources of the specified type that you are authorized to access.

CEDf
When EDF is active, CEDF initiation and termination screens are to be shown by CEDF while this program is running. Other screens are also to be shown unless the program was translated using the NOEDF translator option.

CLass (*value***)**
specifies the 2-character suffix of a program list table (PLT). Use this option to inquire on those programs defined in the specified PLT. Note that, unlike a shutdown PLT, it is not mandatory to define a program definition for a startup PLT. However, if you want to use the CLASS option, you must define and install a program resource definition for the PLT specified by “clasid”.

If the clasid specified is the suffix of a PLT that does not have a program definition, the INQUIRE PROGRAM CLASS(clasid) command returns a ‘CLASS NOT FOUND’ message.

You cannot specify CLASS and a program name.

DIsabled
The program is not available for use. Programs beginning with “DFH” cannot be disabled because these characters are reserved for use by CICS.

DPlsubset
The program is restricted to the DPL API subset, as for a distributed program link request, when it runs in the local CICS region.

A program is always restricted to the DPL subset when it is invoked in a remote region via a DPL request, regardless of this option.

Enabled
The program is available for use.

Fullapi
The program is not restricted to the DPL subset of the CICS API when it runs in the local CICS region, and can use the full API.

JVM
The program is to run in a Java Virtual Machine (JVM).

JVMClass (*value***)**
specifies the name of the main class in the Java program that is to be given control by the JVM.

JVMProfile (*value***)**
specifies the 8–character name of a JVM profile that is to be used for the JVM

in which this Java program runs. Any instances of this program that are currently running in a JVM with the old JVM profile are unaffected, and are allowed to finish running.

When you use the name of a JVM profile anywhere in CICS, you must enter it using the same combination of upper and lower case characters that is present in the z/OS UNIX file name. If you need to enter the name of a JVM profile in mixed case when you are using the CEMT transaction, ensure that the terminal you use is correctly configured, with upper case translation suppressed.

NEwcopy
CICS is to use a new copy of the program when the program ceases to be in use by any transaction. (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.) It is possible for CICS to replace the program with the new version during a single transaction, at a point when one use of the program has completed, and a subsequent use has yet to start.

CICS loads the new version either from the DFHRPL or dynamic LIBRARY concatenation, or uses an LPA-resident version, depending on the PRIVATE or SHARED options. PRIVATE is the default setting.

Note: NEWCOPY cannot be specified for a program specified with the HOLD option, or for a Java program that runs in a JVM. Use the PERFORM JVMPOOL TERMINATE command to refresh Java programs.

NOJvm
The program is not to run in a JVM. Any value in the JVMCLASS option is ignored and the runtime environment of the program is unknown until it is next loaded by CICS, at which point its runtime environment is determined.

NOCedf
All CEDF activities, including initiation and termination screens, are to stop while this program is being processed.

PHasein
CICS is to use a new copy of the program now for all new transaction requests. CICS continues to use the old copy for all currently running transactions until they have finished (RESCOUNT equal to zero). CICS loads the new version either from the DFHRPL or dynamic LIBRARY concatenation, or uses an LPA-resident version, depending on the PRIVATE or SHARED options. PRIVATE is the default setting.

Note: PHASEIN cannot be specified for a program specified with the HOLD option, or for a Java program that runs in a JVM. Use the PERFORM JVMPOOL TERMINATE command to refresh Java programs.

PHASEIN performs a REFRESH PROGRAM function to inform the loader domain that a new version of the program has been cataloged and that this version of the named program should be used in all future ACQUIRE requests.

Until the program goes to a zero RESCOUNT and a subsequent ACQUIRE PROGRAM has been performed, CEMT INQUIRE PROGRAM ( ) will return information on the first copy of the program.

PRIvate
The next new copy of the program is to be loaded from the DFHRPL or dynamic LIBRARY concatenation. This is the default action.

Shared
CICS is to use a shared copy of the program from the LPA as the next new copy of the program. This value is ignored for a Java program that runs in a JVM.

CEMT SET QUEUE

Comments

See “CEMT SET TDQUEUE” on page 449. (QUEUE is accepted for INQUIRE commands, but the correct keyword is TDQUEUE.)

CEMT SET STATISTICS

Change statistics attributes.

Description

SET STATISTICS allows you to change some of the values that control the accumulation and recording of statistics.

For information about printing statistics using the CICS-supplied utility program DFHSTUP, see the *CICS Operations and Utilities Guide*.

NEXTTIME cannot be specified on an INQUIRE or SET STATISTICS command, but its value is displayed when INQUIRE STATISTICS is issued. NEXTTIME indicates the time at which statistics will next be recorded and reset. If SET STATISTICS is in the OFF condition, this is at the end of the day. Otherwise, in the ON condition, it is the earlier time of either the next interval to occur or the end of the day.

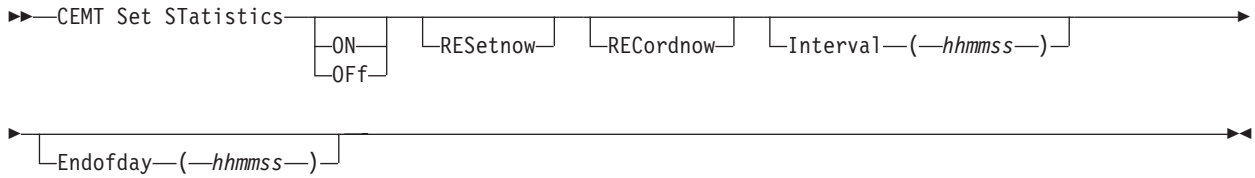
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET STATISTICS` (the minimum abbreviation is `CEMT S ST`). You get a display that lists the current status, similar to that obtained by CEMT INQUIRE STATISTICS. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET STATISTICS` (`CEMT S ST`) followed by one or more attribute settings that you wish to change. For example, `cemt s st on res` will switch CICS statistics ON and will reset the statistics counters.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET STATISTICS



Options

Endofday (*hhmmss*)
specifies the end-of-day time for recording statistics. At end-of-day, the current values of the CICS statistics counters are written out to an SMF data set and the statistics counters are reset.

End-of-day times must be in the range 000000-235959. The default end-of-day time is 12 midnight (000000).

Interval (*hhmmss*)
specifies the length of time during which the statistics counters are incremented. At the end of each interval, the accumulated statistics are recorded and the statistics counters are reset.

Interval times must be in the range 000100-240000. The default interval value is 030000. These interval settings only take place if you set the statistics recording status to ON. You can set the statistics recording status ON or OFF using the RECORDING option on this command or by using the STATRCD system initialization parameter.

The arrival of the ENDOFDAY time always causes the current interval to be ended (possibly prematurely) and a new interval to be started. Only end-of-day statistics are recorded at the end-of-day time, even if it coincides exactly with the expiry of an interval.

Changing the ENDOFDAY value has an immediate effect on the times at which INTERVAL statistics are recorded. New interval expiry times are recalibrated in accordance with the new end-of-day time.

When you change the INTERVAL value or the ENDOFDAY value (and also when CICS is initialized), the length of the current (or first) interval is adjusted so that it expires after an integral number of intervals from the end-of-day time.

These rules are illustrated by the following example, where **I** indicates an interval recording, INTERVAL is set at 3-hourly intervals, CICS is initialized at 0815 hours, and **E** indicates the ENDOFDAY setting at 1700 hours.

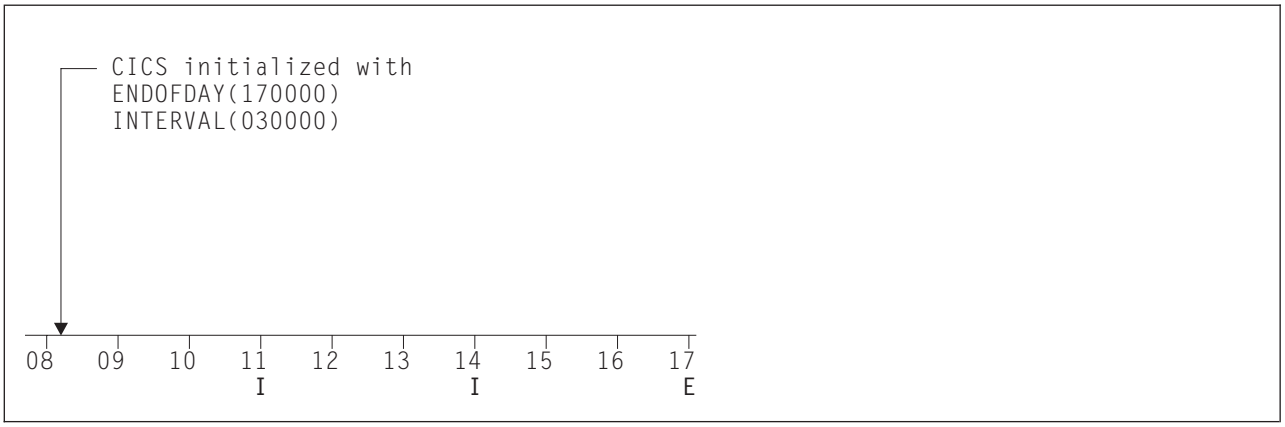


Figure 176. Changing the INTERVAL or ENDOFDAY value

Note: In this example, there would have been an interval at 0800 hours if CICS had been initialized earlier.

If you want your end-of-day recordings to cover 24 hours, set INTERVAL to 240000.

OFf
The recording of interval statistics is switched OFF. (End-of-day, requested, and unsolicited statistics are always recorded.) Counters are still incremented as usual and the recording of end-of-day, requested, and unsolicited statistics is still made. Statistics can be recorded at any time by issuing a CEMT PERFORM STATISTICS command.

ON
The recording of interval statistics is switched ON. Counters are incremented as usual, and are reset at the expiry of each interval and at end-of-day. Statistics can be recorded at any time by issuing a CEMT PERFORM STATISTICS command.

RECordnow
The existing statistics are to be written to an SMF data set. The statistics that you are given are those collected since the last ENDOFDAY or INTERVAL collection, or since the last time the RESETNOW function was used in this or the PERFORM STATISTICS command. RECORDNOW RESETNOW only takes effect when the recording status is changed.

Note: RECORDNOW RESETNOW both resets the statistics counter and also writes the existing statistics to an SMF data set. The statistics that you are given are those collected since the last ENDOFDAY or INTERVAL collection, or since the last time the RESETNOW function was used in this or the PERFORM STATISTICS command. RECORDNOW RESETNOW only takes effect when the recording status is changed.

RESetnow
The statistics counters are to be reset. The reset takes effect only if the recording option is changed from OFF to ON or from ON to OFF. For information about the effect of reset on each individual counter, see the CICS statistics tables in the *CICS Performance Guide*.

The action, for each counter, is one of the following:

- Not reset
- Reset to zero
- Reset to 1
- Reset to current value (this applies to peak values)
- An exception to the above.

CEMT SET SYDUMPCODE

Change the attributes of the system dump codes.

Description

SET SYDUMPCODE allows you to define the entries that you require in the system dump table. You control system dumps by creating an entry in the system dump table for each dump code that requires a change from the default action.

A system dump code is the CICS message number with the DFH prefix removed. Typically this leaves a 6-character code comprising 2 alphabetic characters and 4 numeric characters, for example AP0001. The command syntax permits up to 8 characters for the dump code, and the dump codes you specify are padded with trailing spaces.

Each message has its own system default actions that you cannot necessarily override with the SET command. For example, when a serious error occurs, CICS shuts down even if you have asked for no shut down. See *CICS Messages and Codes* for information about the system action for each message.

There are other system parameters that take priority over your dump table entries. For example, EXEC CICS SET SYSTEM NOSYSDUMP, or a DUMP=NO system initialization parameter, inhibits system dumping. For more information on using dumps, see the *CICS Problem Determination Guide*.

The transaction dump command SET TRDUMPCODE can also cause CICS to produce system dumps for the specified transaction dump code.

The SET command allows you to:

- Add a new entry to the system dump table
- Remove an entry from the system dump table
- Indicate whether or not you want a system dump taken
- Indicate whether the system dump request is to be local or related
- Indicate whether or not you want CICS to shut down following the dump
- Set the maximum number of dumps
- Reset the counter of the current number of dumps to zero
- Determine whether or not a dump produced by CICS is eligible for any suppression by the MVS Dump Analysis and Elimination (DAE) component.

Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET SYDUMPCODE` (the minimum abbreviation is `CEMT S SYD`) with code or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE SYDUMPCODE. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET SYDUMPCODE` (`CEMT S SYD`) with code or ALL, followed by one or more attribute settings that you wish to change. For example, to add a system dump code:

  ```
  CEMT SET SYDUMPCODE(value) ADD
  ```

  To remove a system dump code:

  ```
  CEMT SET SYDUMPCODE(value) REMOVE
  ```

  To change the maximum count to 10:

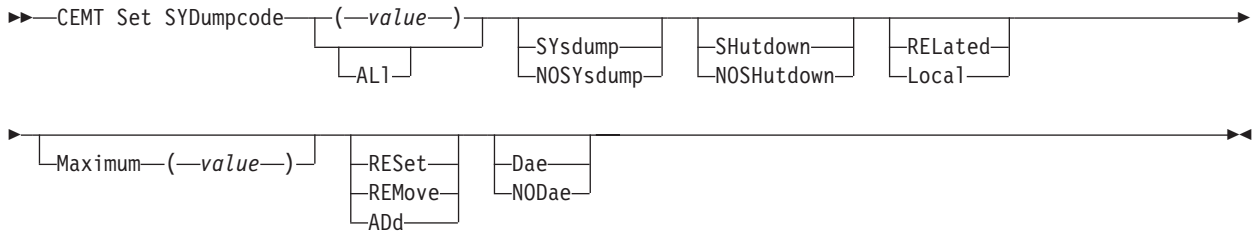
  ```
  CEMT SET SYDUMPCODE(value) MAX(10)
  ```

  Shut down CICS on occurrence of this system dump code:

  ```
  CEMT SET SYDUMPCODE(value) SHUTDOWN
  ```

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET SYDUMPCODE



Options

(*value***)**
specifies an 8-character system dump code. See *CICS Messages and Codes* for details of the codes.

ADd
This system dump code is added to the system dump code table.

ALl
specifies that any changes you request are made to all resources of the specified type that you are authorized to access.

Dae
A system dump is eligible for suppression by the DAE component.

Local
A system dump request is not to be sent to other MVS images in the sysplex for this system dump code.

Maximum (*value***)**
specifies the maximum number of dump calls for this system dump code that result in system dumps being taken. The value must be in the range 0-999; a value of 999 means the default, 'no limit'.

NODae
A system dump will not be suppressed by the DAE component. (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 *OS/390 MVS Diagnosis: Tools and Service Aids* manula.

NOSHutdown
CICS is not to be shut down after the occurrence of an error corresponding to this dump code.

NOSYsdump
A system dump is not to be taken for this system dump code.

RELated
A system dump request is to be sent to MVS images in the sysplex which are running XCF/MRO connected CICS regions related to the CICS region on which the dump is initiated.

Note: A related CICS region is one in which the unit of work identifiers, in the form of APPC tokens, of one or more tasks match those in the CICS region that issued the dump request.

REMove
This system dump code is removed from the system dump code table.

RESet
The current number of calls for this dump code is reset to zero.

SHutdown
CICS is to be shut down.

SYsdump
A system dump is to be taken for this system dump code.

CEMT SET SYSTEM

Change the system attributes.

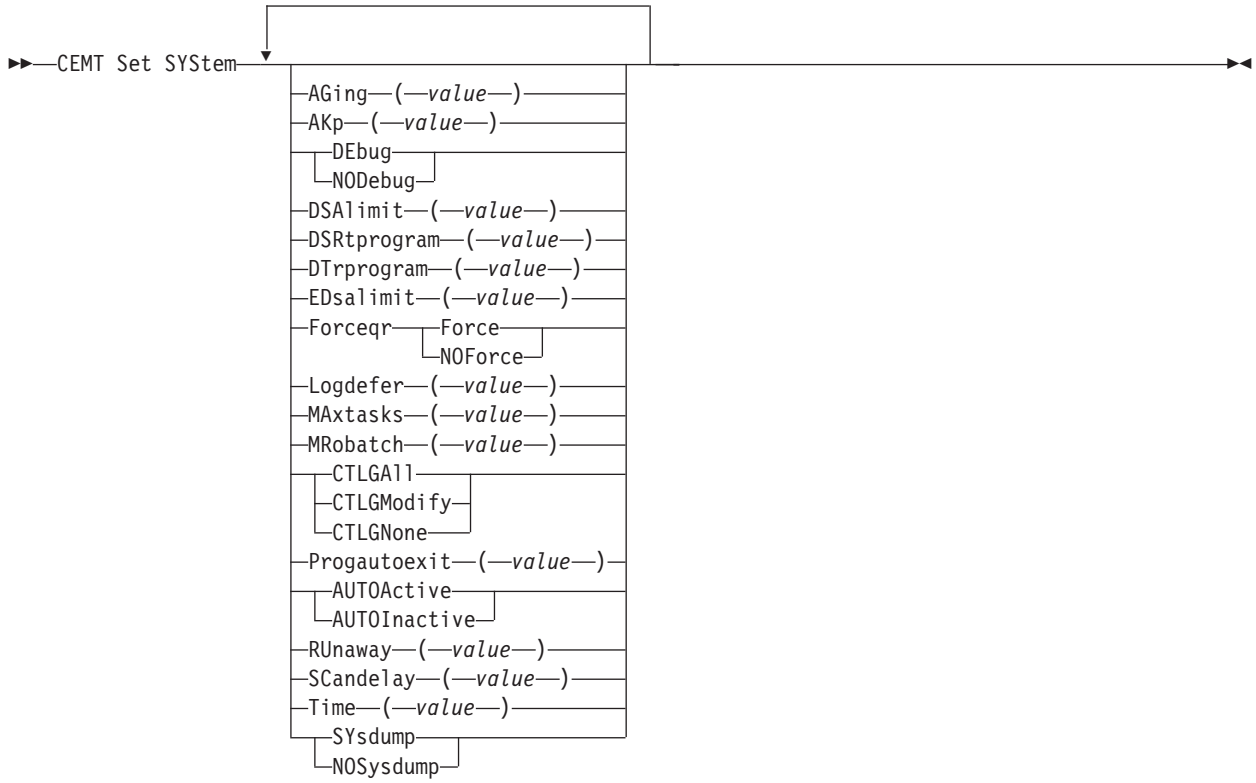
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET`. The SYSTEM keyword does not have to be typed. If you type just `CEMT SET`, SYSTEM is assumed as the default. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE SYSTEM. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET` followed by one or more attribute settings that you wish to change. For example, `cemt s ag(`*value*`) ma(`*value*`)` resets the values for priority aging and the maximum number of tasks allowed at any one time in the CICS system.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET SYSTEM



Options

AGing (*value*)
specifies the time factor for CICS to use in the priority aging algorithm for incrementing the priority of a task. The value is expressed as “milliseconds per unit of priority”, and must be in the range 00000–65535. The priority aging factor is defined initially by the PRTYAGE system initialization parameter, in which the value represents the number of milliseconds that must elapse before the priority of a waiting task can be adjusted upward by 1. See the *CICS System Definition Guide* for information about the PRTYAGE parameter.

AKp (*value*)
specifies 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. It must have a value in the range 00200–65535. You cannot change this value if AKPFREQ=0 was specified at system initialization.

AUTOActive
Autoinstall for programs is active. On first use, if a program, mapset, or partitionset is not defined, the definition is created dynamically.

AUTOInactive
Autoinstall is not active. If a program is not defined, a PGMIDERR or transaction abend occurs when it is referenced.

CTLGAll
All autoinstalled program definitions are to be cataloged and restored on a warm or emergency restart.

CTLGModify
Autoinstalled program definitions are to be cataloged only if they are modified (for example, by a CEMT SET PROGRAM command), so that the modified definitions are restored on a warm or emergency restart.

CTLGNone
No autoinstalled program definitions are to be cataloged. They are autoinstalled again after a warm or emergency restart.

DEbug
Specifies that you want to use CICS 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.

For more information about debugging profiles, see the *CICS Application Programming Guide*.

DSAlimit (*value***)**
specifies the maximum amount of storage, as a total number of bytes, within which CICS can dynamically allocate storage for the four individual DSAs that reside below the 16MB boundary. See the *CICS System Definition Guide* for a description of the DSALIM system initialization parameter.) 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 dynamic storage is freed in the individual DSAs.

Note: Dynamic changes to the DSA limit are cataloged in the *local* catalog, and override values specified in the system initialization table during all forms of restart-initial, old, and warm. The cataloged value is *not* used if:

- You specify startup values as system initialization parameters overrides (for example, in SYSIN).
- You re-initialize the CICS catalog data sets.

DSRtprogram (*value***)**
specifies the name of the distributed routing program.

DTrprogram (*value***)**
specifies the name of the dynamic routing program.

EDsalimit (*value***)**
specifies the maximum amount of storage, as a total number of bytes, within which CICS can dynamically allocate storage for the four individual DSAs that reside above the 16MB boundary. See the *CICS System Definition Guide* for a description of the EDSALIM system initialization parameter.) 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 dynamic storage is freed in the individual DSAs.

Note: Dynamic changes to the EDSA limit are cataloged in the *local* catalog, and override values specified in the system initialization table during all forms of restart-initial, old, and warm. The cataloged value is *not* used if:

- You specify startup values as system initialization parameters overrides (for example, in SYSIN).
- You re-initialize the CICS catalog data sets.

Force
Force all CICSAPI user application programs to run under the QR TCB, even if they are defined with the CONCURRENY(THREADSAFE) attribute. Force does not apply to OPENAPI programs or C/C++ programs compiled with XPLINK.

Logdefer (*value***)**
specifies the log deferral interval in milliseconds, 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. It is in the range 0-65535. See the *CICS System Definition Guide* for information about the Logdefer parameter and associated SIT parameter LGDFINT.

MAxtasks (*value***)**
specifies the maximum number of tasks, both active and suspended, allowed at any one time in the CICS system. It must have a value in the range 1–999.

Note: The value assigned to MAXTASKS might be less than the requested value, because of CICS storage constraints. If this occurs, the message CEILING REACHED is displayed when the request is made.

MRobatch (*value***)**
specifies the number of MRO requests from connected regions that are to be batched before this region is posted. It must have a value in the range 001–255.

NODebug
Specifies that you do not want to use CICS 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.

For more information about debugging profiles, see the *CICS Application Programming Guide*.

NOForce
Do not force user application programs to run under the QR TCB. CICS is to honor the CONCURRENCY(THREADSAFE) attribute on user application programs, and allow user programs to run on an open TCB to avoid unnecessary TCB switching.

NOSysdump
specifies that system dumps are to be suppressed.

Progautoexit (*value***)**
specifies the name of the user-provided program that is called by the program autoinstall code to select or modify a model definition.

RUnaway (*value***)**
specifies the interval, in milliseconds, for which a task can have control before it is assumed to be looping.

You can specify 0, or a value in the range 500–2700000. The value chosen is rounded down to a multiple of 500. CICS purges a task if it has not given up control after this interval (that is, if the task appears to be looping). If you specify zero, runaway task control is inoperative (that is, tasks do not get purged if they appear to be looping). The value is independent of, and can be less than, the value in the TIME operand.

CICS runaway-task detection is based upon task time, that is, the interval is decremented only when the task has control of the processor. You do not, therefore, need to allow for the effect of other jobs when setting the value.

SCandelay (*value***)**
specifies the maximum number of milliseconds between the receipt of a VTAM request, and the time when CICS begins to process the request. The number must be in the range 0–5000, and must be less than or equal to the TIME value.

In the summary report of the dispatcher statistics, SCANDELAY is referred to as ICVTSD.

SYsdump
specifies that system dumps are not to be suppressed.

Time (*value***)**
specifies the interval, in milliseconds, for which CICS releases control to the operating system if no transactions are ready to resume processing. This interval is known as the 'region exit interval'.

It can have a value in the range 100–3600000, and must be greater than or equal to the SCANDELAY value.

In the summary report of the dispatcher statistics, TIME is referred to as ICV.

CEMT SET TASK

Change the status of tasks.

Description

You cannot specify a list of identifiers, nor can you use the symbols * and + to specify a family of tasks.

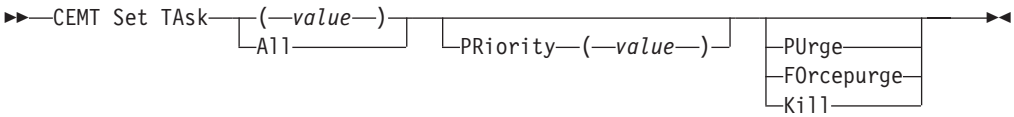
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET TASK` (the minimum abbreviation is `CEMT S TA`) with (*value*) or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE TASK. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET TASK` (`CEMT S TA`) with number or ALL, followed by one or more attribute settings that you wish to change. For example, `cemt s ta(`*value*`) pu` will make the specified task purgeable in system-stall conditions.

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET TASK



Options

(*value***)**
The CICS-generated task number. This is in the range 1–99999.

All
Any changes you request are made to all resources of the specified type that you are authorized to access.

FOrcepurge
The task is to be terminated as soon as it is consistent with system integrity and without regard to data integrity. CICS cannot always determine whether a forced purge is safe; it is possible to abend the system when you specify this option. For example, if a task is forcepurged during backout processing, CICS terminates abnormally. If you want to terminate a task but do not want to terminate CICS, you should use PURGE instead of FORCEPURGE.

Kill
The task is to be terminated. System and data integrity is not guaranteed. The KILL option extends the PURGE and FORCEPURGE options. It should be used only after an attempt has been made to PURGE or FORCEPURGE a task. The KILL option does not guarantee integrity of any kind, but in some situations it allows the user to free up a stalled region enabling the region to continue processing. In some cases, for example, if a task is killed during backout processing, CICS terminates abnormally.

PRiority (*value***)**
specifies the priority of the task, in the range 0–255 where 255 is the highest priority.

PUrge
The task is terminated as soon as both system and data integrity can be maintained.

If the task is in any way associated with an open TCB (for example, it involves a Java program running in a JVM on a J8 or J9 mode TCB), you could experience a delay of about a minute before the task is finally purged.

CEMT SET TCLASS

Reset the maximum number of tasks and the purge threshold for a transaction class.

Syntax

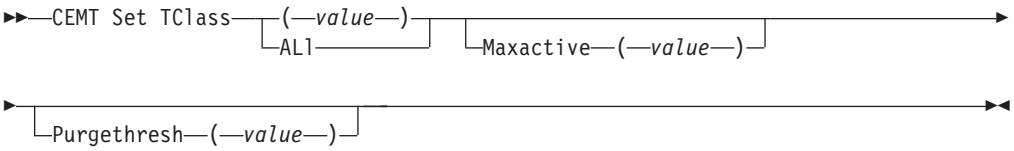
Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET TCLASS` (the minimum abbreviation is `CEMT S TC`) with (*value*) or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE TCLASS. You can then tab to the highlighted or blank fields and overtype them with the required values.

- Type CEMT SET TCLASS (CEMT S TC) followed by one or more attribute settings that you wish to change. For example, cemt s tc(*value*) p(3) will reset the purge threshold value to 3.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET TCLASS



Options

(*value***)**
specifies the 8-character transaction class name. If no transaction class name is assigned, it defaults to DFHTCL00.

ALl
specifies that any changes you request are made to all resources of the specified type that you are authorized to access.

Maxactive (*value***)**
specifies the largest number of transactions in the transaction class which are allowed to run concurrently. The value can be in the range 0-999.

Purgethresh (*value***)**
specifies the limit at which the queuing transactions are purged for the transaction class. It can have a value in the range 0–1000000. Zero means that the transactions are not purged. A nonzero number “n” means that the first (n-1) transactions to arrive are queued and the nth is purged.

CEMT SET TCPIP

Open or Close CICS TCP/IP Service.

Description

SET TCPIP allows you to open or close CICS internal sockets support.

Note: This command has no effect on the sockets support provided by the TCP/IP for OS/390 CICS Sockets Feature.

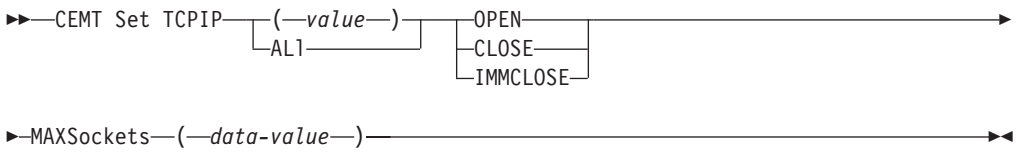
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type CEMT SET TCPIP (the minimum abbreviation is CEMT S TCPIP). You get a display that lists the current status, similar to that obtained by CEMT INQUIRE TCPIP. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type CEMT SET TCPIP (CEMT S TCPIP) with (*value*) or ALL, followed by one or more attribute settings that you wish to change. For example, cemt s tcpip *status* open opens CICS internal sockets support.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET TCPIP



Options

Openstatus
specifies whether or not TCPIP is to be opened (that is, able to process new incoming work, and complete ongoing work), and if TCP/IP support is to be closed, how the close should be done. CVDA values are:

OPEN CICS internal sockets support is to be opened.

Note: If you are using CEMT SET TCPIP OPEN to initialize CICS sockets, you must also set TCPIP=YES at CICS initialization by SIT or a SIT override. Otherwise, 'NOT IN SYSTEM' is returned.

CLOSE
CICS internal sockets 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 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.

MAXSockets(*data-value***)**
specifies 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 use the SET SYSTEM command to specify a value greater than that in MAXFILEPROC, CICS resets the limit to MAXFILEPROC. The message "MAXFILEPROC LIMIT REACHED" is displayed when the request is made.

Note that sockets created by Java programs running on threads that are not managed by CICS do not count towards the 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.

CEMT SET TCPIPSERVICE

Modify the status of a service using CICS internal TCP/IP support.

Description

SET TCPIPSERVICE allows you to update the information relating to a service which uses CICS internal TCP/IP support.

Note: This command has no effect on the sockets support provided by the TCP/IP for CICS Sockets Feature.

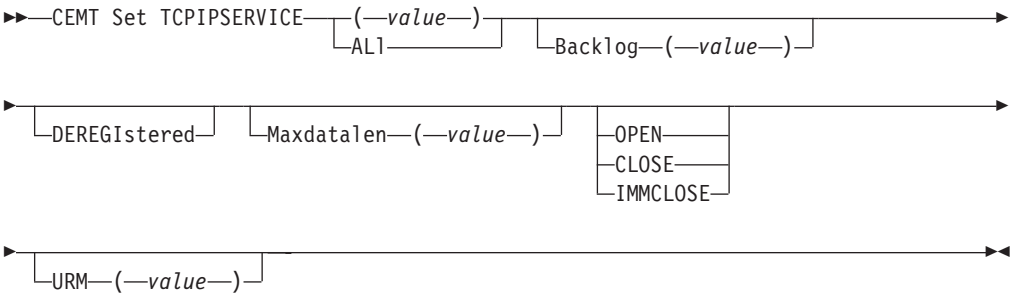
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET TCPIPSERVICE` (the minimum abbreviation is `CEMT S TCPIPS`). You get a display that lists the current status, similar to that obtained by CEMT INQUIRE TCPIPSERVICE. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET TCPIPS` (`CEMT S TCPIPS`) with (*value*) or ALL, followed by one or more attribute settings that you wish to change. For example, `cemt s tcpips status open` opens CICS internal sockets support.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET TCPIPSERVICE



Options

BACKLOG
Changes the maximum number of requests which 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 current state of Workload Manager (WLM) Domain Name System (DNS) registration of this TCPIPSERVICE. 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 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.

MAXDATALEN
Changes the maximum length of data that may be received by CICS as an HTTP server. This can be 3K through 524288K. The default is 32K.

OPENSTATUS
Changes the status of the service. CVDA values are:

OPEN CICS internal TCP/IP sockets support is to start listening for incoming requests on the specified port.

CLOSE
CICS internal TCP/IP sockets support is to stop listening for incoming work on the specified port. All active tasks using this service are allowed to complete.

IMMCLOSE
CICS internal sockets support is to stop listening for incoming work on the specified port immediately. Any active tasks using this service are terminated abnormally.

URM
Specifies the 8-character name of the program to be used as the Service User-replaceable module by the attached task. You can specify either an installation specific program or the CICS-supplied default for the service.

CEMT SET TDQUEUE

Changes the attributes of transient data queues.

Description

The **SET TDQUEUE** command changes some of the attributes of a transient data queue. The queue must not be REMOTE or INDIRECT.

You cannot alter the following attributes unless the queue is fully disabled:

- ATIFACILITY
- ATITERMID
- ATITRANID

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 TRIGGERLEVEL, OPENSTATUS, and ENABLESTATUS attributes may be altered regardless of whether the queue is enabled or disabled. The value of the ENABLESTATUS attribute 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 nonrecoverable intrapartition queue and an attempt is made to disable the queue, the queue enters a "disable pending" state. The last task to use the extrapartition queue fully disables it.

If an attempt is made to disable a logically recoverable intrapartition TD queue when there are UOWs enqueued upon it, the queue enters a "disable pending" state. The last UOW to obtain the enqueue fully disables the queue. If a UOW has updated a logically recoverable queue and suffers an indoubt failure, the queue cannot be disabled until the indoubt failure has been resolved.

If a UOW owns an enqueue on a queue that is in a "disable pending" state, it is allowed to continue making updates.

When a queue is in a "disable pending" state, no new tasks can alter the queue's state or its contents. A disabled response is returned when a READQ, WRITEQ, or DELETEQ request is issued against a destination that is in a "disable pending" state.

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) in 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**.

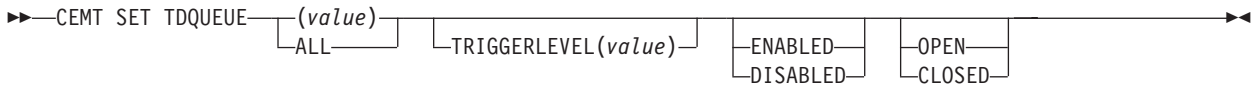
Syntax

Press the Clear key to clear the screen. There are two ways of starting this transaction:

- Type `CEMT SET TDQUEUE` (the minimum abbreviation is `CEMT S TD`) with `destid` or `ALL`. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE TDQUEUE. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET TDQUEUE (CEMT S TD)` with `(value)` or `ALL`, followed by one or more attribute settings that you wish to change. For example, `cemt s td(value) en op` will make a named extrapartition queue accessible by applications and open.

Typing `?` at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET TDQUEUE



Options

(value)
The 1- to-4 character identifier of a transient data queue.

ALL
Specifies that any changes you request are made to all resources of the specified type that you are authorized to access.

CLOSED
The queue is closed.

DISABLED
The queue cannot be accessed by applications, although it can still be open. A queue that has suffered an indoubt failure cannot be disabled.

ENABLED
The queue can be accessed by applications.

OPEN
The extrapartition queue is open.

TRIGGERLEVEL *(value)*
Specifies the number, 0-32767 of requests for output to a queue that there must be before automatic transaction initiation (ATI) occurs. Specifying zero disables ATI.

CEMT SET TERMINAL

Change the attributes of named terminals.

Description

The SET TERMINAL command changes some of the values of a named terminal defined in the terminal control table (TCT). It cannot be used for APPC sessions.

In an SNA environment, a terminal is the CICS representation of an SNA logical unit with which CICS can be in communication. (SNA environment means that CICS is using VTAM as the access method for communication with SNA logical units.)

This logical unit can be:

- A physical terminal (such as a 3277)
- A function of a cluster controller (such as a 3790 Type 2 batch function)
- An intersystem communication (ISC) or interregion communication (IRC) session that has been generated by the CEDA transaction.

If you are familiar with network names, you may prefer to use the command that uses these names. See “CEMT SET NETNAME” on page 427.

The message “TEMP OUT SERVICE” can be displayed if recovery is in progress for the terminal. The ”TEMP OUT SERVICE” indicator is reset by simply refreshing the screen, or by overtyping the “INS” field with “INS” or “OUT”. If “TEMP OUT SERVICE” continues to be displayed, the cause of the indicator should be investigated.

Syntax

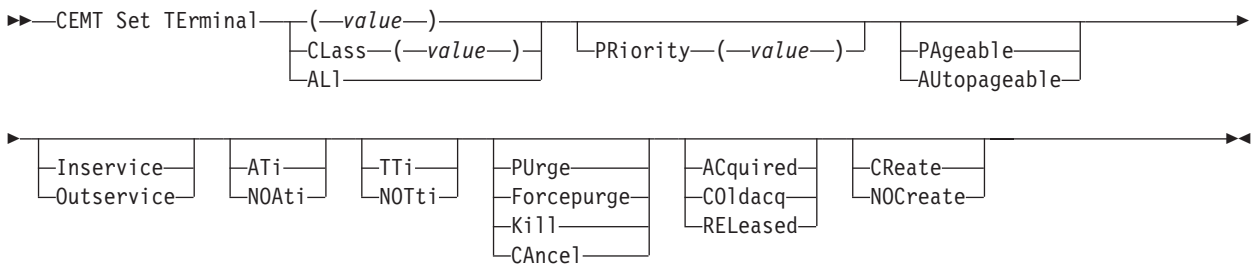
Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET TERMINAL` (the minimum abbreviation is `CEMT S TE`) with (*value*), CLASS(*value*), or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE TERMINAL. You can then tab to the highlighted or blank fields and overtype them with the required values.

- Type CEMT SET TERMINAL (CEMT S TE) with (*value*), CLASS(*value*), or ALL, followed by one or more attribute settings that you wish to change. For example, cemt s te(*value*) i cr specifies that a named terminal is available for use (inservice) and that it can be acquired automatically by ATI transactions (create).

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET TERMINAL



Options

(*value***)**
A terminal identifier (1-4 characters) as specified in an installed terminal definition.

If the terminal name is, for example, S201, this option is coded thus:

```
CEMT SET TERMINAL(S201)
```

ACquired (*vtam only***)**
CICS is in session with the logical unit represented by the terminal.

ALl
Any changes you request are made to all resources of the specified type that you are authorized to access.

ATi (*vtam only***)**
The terminal is available for use by transactions that are initiated automatically from within CICS or, if the terminal is an ISC session, by transactions that are using this session as an alternative facility to communicate with another system.

AUtopageable (*vtam only***)**
Pages after the first in a series are to be written to the terminal automatically.

CAncel
AIDs queuing for the specified terminal are canceled. AIDs representing scheduled and allocated requests waiting in the local CICS system for the specified terminal are canceled. However, 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.

An 'AIDS CANCELED' message appears on the CEMT panel whenever AIDs are deleted using the CANCEL option of the CEMT SET TERMINAL command.

CLass (*value***)**
The 1-or 2-character suffix of a terminal list table (TLT).

COldacq (*vtam only***)**
This is a special form of ACQUIRED, where no resynchronization is required. If the previous session abended, the use of COLDACQ overrides CICS integrity control. This could lead to integrity problems. Also, you should check the CSMT log for an activity keypoint after the restart of a session following a CICS failure. If there is no activity keypoint, you should issue COLDACQ again after the next emergency restart.

CReate (*vtam only***)**
If the terminal is not in session, CICS is to acquire it if it is needed to satisfy an ATI request.

Forcepurge
Any transaction running with this terminal is immediately terminated abnormally. Data integrity is not guaranteed. In some extreme cases (for example, if an error occurs during backout processing), CICS might terminate abnormally.

Inservice
The terminal is available for use. For VTAM, INSERVICE means that the terminal can be ACQUIRED.

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.

NOAti (*vtam only***)**
The terminal is not available for use by transactions that are initiated automatically from within CICS or, if the terminal is an ISC session, by transactions that are using this session as an alternative facility to communicate with another system.

Note: A terminal cannot be defined with both NOATI and NOTTI.

NOCreate (*vtam only***)**
If the terminal is not in session, CICS is not to acquire it to satisfy an ATI request. A session must be started by, for example, a logon request or a CEMT SET TERMINAL ACQUIRED command before the ATI request can be satisfied.

If NOCREATE is set for an LU6.1 ISC session and there are no allocatable sessions left, the connection is placed OUTSERVICE.

NOTti (*vtam only***)**
This terminal cannot be used by transactions that are initiated from this terminal.

Note: A terminal cannot be defined with both NOATI and NOTTI.

Outservice
The terminal is not available for use. Setting a terminal OUTSERVICE means that the terminal can no longer be used by transactions. If PURGE or FORCEPURGE is also specified, any transaction using the terminal is terminated abnormally. If PURGE or FORCEPURGE is not specified, the transaction is allowed to terminate normally, but no further transactions are

allowed to use the terminal. For VTAM, setting a terminal OUTSERVICE also causes it to be released and the operator to be signed off, either immediately or when the current transaction has terminated.

In an LU6.1 ISC session the connection is set OUTSERVICE if there are no allocatable sessions left.

To set an IRC session OUTSERVICE, you must set the connection to the MRO partner OUTSERVICE.

PAgeable (*vtam only***)**
Pages after the first in a series are to be written to the terminal only if you request them to be.

PRiority (*value) (vtam only***)**
specifies the priority of a terminal relative to other terminals. The priority of a task is the sum of the transaction priority, the terminal priority, and the operator priority. Priority has no meaning for terminals that are ISC sessions being used as alternative facilities. The value must be in the range 0–255, where 255 is the highest priority.

PUrge
Any transaction running with this terminal is purged only if system and data integrity can be maintained.

RELeased (*vtam only***)**
CICS is not in session with the logical unit represented by the terminal. Setting a terminal RELEASED causes the session to be terminated. Running transactions are allowed to finish unless PURGE or FORCEPURGE is also specified.

TTi (*vtam only***)**
This terminal can be used by transactions.

CEMT SET TRANSACTION

Change some of the attributes of a selected transaction.

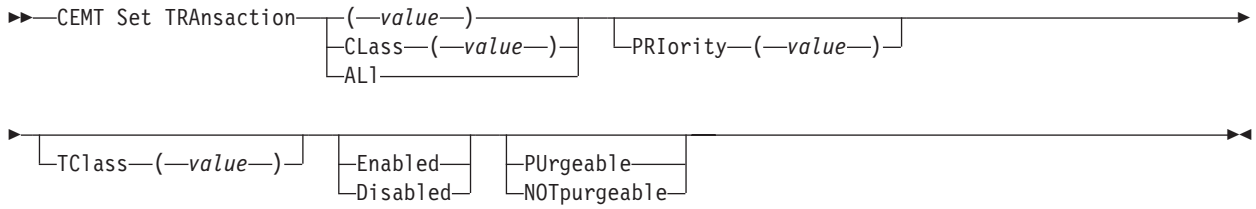
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET TRANSACTION` (the minimum abbreviation is `CEMT S TRANS`) with (*value*), CLASS(*value*), or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE TRANSACTION. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET TRANSACTION` (`CEMT S TRANS`) followed by one or more attribute settings that you wish to change. For example, `cemt s trans(`*value*`) e pu` specifies that a named transaction is available for use (enabled) and is system-purgeable.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET TRANSACTION



Options

(*value***)**
specifies a transaction identifier (1-4 characters). Only transactions that have been defined in the CICS system definition (CSD) file and installed on the running CICS system are accessible through CEMT.

ALl
specifies that any changes you request are made to all resources of the specified type that you are authorized to access.

CLass (*value***)**
specifies the 2-character suffix of a transaction list table (XLT).

Disabled
The transaction is not available for use. Transactions that have identifiers beginning with "C" cannot be disabled because these are reserved for use by CICS.

When the CEMT SET TRANSACTION DISABLED command is used, existing transactions run to completion before being DISABLED.

Note: If a transaction is disabled, this does not prevent a START command which names this transaction from being shipped to a remote region. When a task is attached for the requested transaction, CICS checks that the transaction is enabled in the remote region.

Enabled
The transaction is available for use.

NOTpurgeable
The transaction cannot be purged.

PRIority (*value***)**
specifies the priority of a transaction relative to other transactions. When a transaction is running as a CICS task, the priority of a task is the sum of the transaction priority, the terminal priority, and the operator priority. The "value" must be in the range 0–255, where 255 is the highest priority.

PUrgeable
The transaction is system-purgeable. This operand relates to the SPURGE parameter on the transaction resource definition and indicates that CICS can purge the transaction in a deadlock time-out situation. See in the *CICS Resource Definition Guide* for information about the SPURGE and DTIMEOUT parameters on a transaction resource definition.

TClass (*value***)**
specifies the 8-character transaction class name to which the transaction belongs. If the transaction does not belong to a class, DFHTCL00 is returned.

The abbreviation for TCLASS is *Tcl*. To remove a transaction from its TCLASS, the *Tcl* field should be set to DFHTCL00. A series of blanks can be inserted, in which case, DFHTCL00 is returned. An added or changed TClass **must** be defined.

CEMT SET TRDUMPCODE

Change the status of a transaction dump.

Description

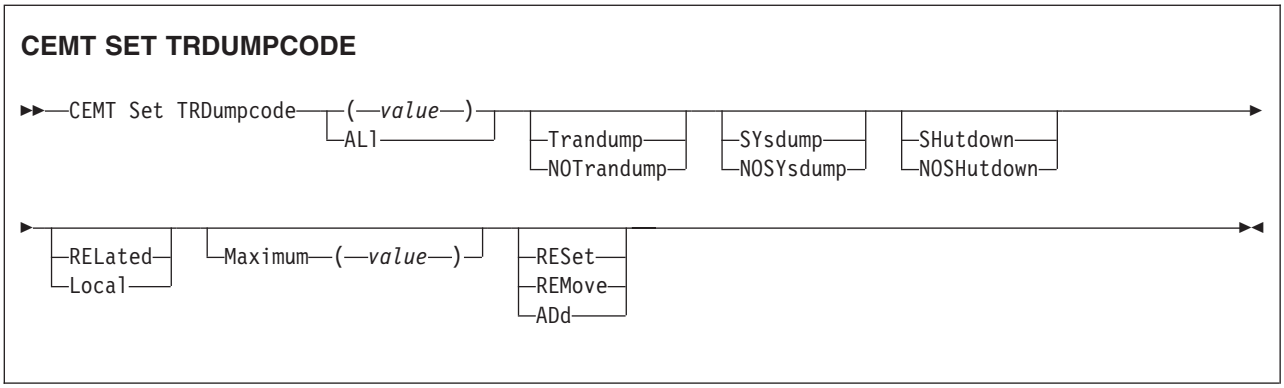
Using the SET function, you can change entries in the transaction dump table to request a transaction dump. You can also set the maximum number of dumps that can be taken. See the *CICS Problem Determination Guide* for information on using dumps.

Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET TRDUMPCODE` (the minimum abbreviation is `CEMT S TRD`) with (*value*) or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE TRDUMPCODE. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET TRDUMPCODE` (`CEMT S TRD`) followed by one or more attribute settings that you wish to change. For example, `cemt s trd(value) t sh` specifies that a transaction dump is to be taken for the named transaction dump code and that the CICS system is to shut down after the occurrence of an error corresponding to this dump code.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.



Options

(*value***)**
specifies a 4-byte transaction dump code. See *CICS Messages and Codes* for a list of transaction dump codes.

ADd
The specified transaction dump code is to be added to the dump code table.

ALl
specifies that any changes you request are made to all resources of the specified type that you are authorized to access.

Local
A system dump request is not to be sent to other MVS images in the sysplex which are running XCF/MRO connected CICS regions related to the CICS region on which the dump is initiated.

Note: A related CICS region is one in which the unit of work identifiers, in the form of APPC tokens, of one or more tasks match those in the CICS region that issued the dump request.

Maximum (*value***)**
specifies the largest number of dump calls for this dump code that result in dumps being taken. The range is 0–999, though a value of 999 means 'no limit'.

NOSHutdown
The CICS system is not to shut down.

NOSYsdump
A system dump is not to be taken.

NOTrandump
A transaction dump is not to be taken.

RELated
A system dump request is to be sent to MVS images in the sysplex which are running XCF/MRO connected CICS regions related to the CICS region on which the dump is initiated.

Note: A related CICS region is one in which the unit of work identifiers, in the form of APPC tokens, of one or more tasks match those in the CICS region that issued the dump request.

REMove
This code is to be removed from the dump code table.

RESet
The current count of dump calls for this dump code is to be reset to zero.

SHutdown
The CICS system is to shut down after the occurrence of an error corresponding to this dump code.

SYsdump
A system dump is to be taken.

Trandump
A transaction dump is to be taken.

CEMT SET TSQUEUE

Delete a temporary storage queue (TS queue). This section applies also to the alternative command, **CEMT SET TSQNAME**. Use either to delete temporary storage queues (TS queues) with names up to 8 characters long. Use SET TSQNAME to delete temporary storage queues (TS queues) with names up to 16 characters long.

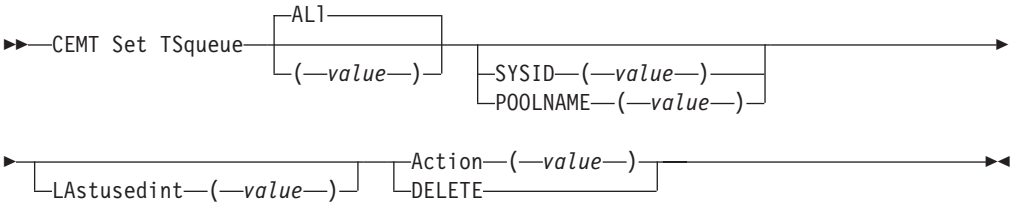
Description

The SET TSQUEUE command enables you to delete a temporary storage queue (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.

Syntax

CEMT SET TSQUEUE



Options

(*value***)**
specifies the identifier of the TS queue

ALl
specifies that any changes you request are made to all resources of the specified type that you are authorized to access.

ACTION(*value***)**
specifies the action to be taken on the queue. The values are:

DELETE
the queue is to be deleted.

LASTUSEDINT
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.

TSQUEUE(*data-value***)**
specifies the 16-character identifier of the TS queue.

CEMT SET UOW

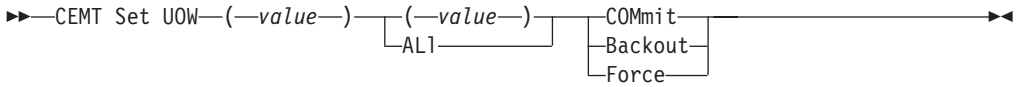
Commit or back out a shunted unit of work.

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.

Syntax

CEMT SET UOW



Options

(*value***)**
specifies the 16-character identifier of the UOW to be committed, backed out, or forced.

ALl
specifies that any changes you request are made to all resources of the specified type that you are authorized to access.

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 backout or commit, as specified on the ACTION option of the TRANSACTION resource definition.

Note that all these values are valid only for UOWs that have been shunted in-doubt. For information about the in-doubt attributes of TRANSACTION definitions, see the *CICS Resource Definition Guide*.

CEMT SET UOWLINK

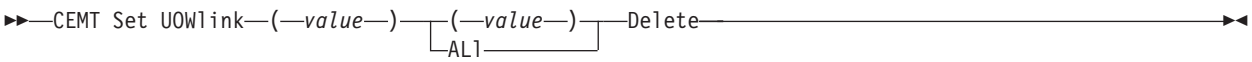
Delete a link to a unit of work that was created by a connection that has since been discarded. UOWLINKS associated with RRS can be deleted when RRS has cold started.

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*..)

Syntax

CEMT SET UOWLINK



Options

(*value***)**
specifies the 4-character identifier of the UOW-connection dependency (the UOW-link) to which this command applies.

ALl
specifies that any changes you request are made to all resources of the specified type that you are authorized to access.

Delete
specifies that the UOW-link is to be deleted.

CEMT SET URIMAP

Enable or disable a URIMAP definition, and apply or remove redirection for a URIMAP definition.

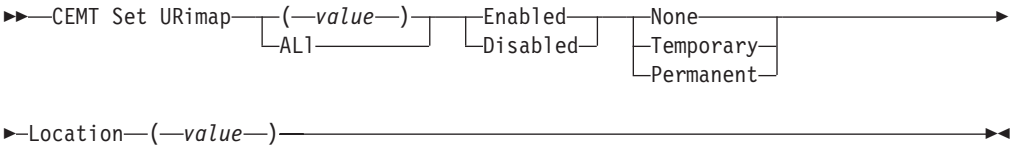
Description

If you disable a URIMAP definition, CICS returns an HTTP response to the Web client with a 503 (Service Unavailable) status code.

You can use the redirection options to 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.

Syntax

CEMT SET URIMAP



Options

(*value***)**
specifies the 8-character name of the URIMAP resource definition to which this command applies.

Enabled
The URIMAP definition can be accessed.

Disabled
The URIMAP definition cannot be accessed.

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).

Location *(value)*
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 checks that the URL is complete and correctly delimited, but CICS does not check that the destination is valid.

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 no redirection is specified, the URL in the LOCATION option is ignored.

CEMT SET VTAM

Open or close the VTAM ACB, set the persistent session delay interval (PSDI), or deregister CICS from membership of a generic resource.

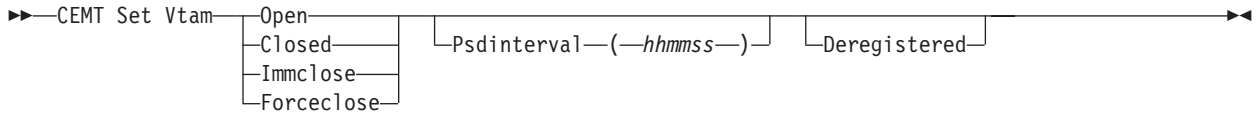
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET VTAM` (the minimum abbreviation is `CEMT S VT`). You get a display that lists the current status, similar to that obtained by CEMT INQUIRE VTAM. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET VTAM` (`CEMT S VT`) followed by one or more attribute settings that you wish to change. For example, `cemt s vt o` specifies that the connection between CICS and VTAM is established.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET VTAM



Options

Closed
The connection between CICS and VTAM is to terminate normally; that is, transactions are allowed to complete and terminals are released normally.

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*.

Forceclose
The connection is terminated immediately, and all terminal connections using the VTAM ACB (access method control block), together with the transactions associated with them, are terminated abnormally. If transactions do not terminate because of their state at the time FORCECLOSE was issued, it is necessary to issue CEMT SET TASK(nn) PURGE or FORCEPURGE before the ACB can be reopened.

The difference between IMMCLOSE and FORCECLOSE is that IMMCLOSE cancels any task at a VTAM terminal and then closes the terminal sessions tidily. FORCECLOSE cancels any task and closes the VTAM ACB, terminating all sessions immediately.

Immclose
The connection between CICS and VTAM is to be terminated immediately. Any transactions using VTAM-connected terminals are terminated abnormally, and all such terminals are lost to CICS.

The difference between IMMCLOSE and FORCECLOSE is that IMMCLOSE cancels any task at a VTAM terminal and then closes the terminal sessions tidily. FORCECLOSE cancels any task and closes the VTAM ACB, terminating all sessions immediately.

Open
The connection between CICS and VTAM is established. With ACF/VTAM, this refers to the opening of the ACF/VTAM ACB through which ACF/VTAM communicates with CICS.

A CEMT SET VTAM OPEN command will fail with a STILL CLOSING message if the command is issued while a previous predatory takeover is still being processed. No attempt is made to open the VTAM ACB, because it is still being closed as a result of the predatory takeover. After the predatory takeover completes, the CEMT SET VTAM OPEN command will work as normal. The failure is notified by a RESP=INVREQ and RESP2=14 condition.

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 (*hhmmss*)
specifies the persistent session delay interval, expressed in the form *hhmmss*. It specifies if and for how long sessions are held in recovery-pending state after CICS fails. The permitted range is 000000 to 235959 inclusive.

000000
If CICS fails, sessions are terminated.

Hhmmss
If CICS fails, sessions are held in recovery-pending state by VTAM for up to the interval specified.

Table 8 shows possible CEMT error messages resulting from unsuccessful attempts to set the persistent session delay interval.

Table 8. CEMT error messages for attempts to set the persistent session delay interval

Message	RESP/RESP2
INVALID PSDINT	RESP=INVREQ, RESP2=4
NOT WITH XRF	RESP=INVREQ, RESP2=8
SETLOGON FAILURE	RESP=INVREQ, RESP2=9

Table 8. CEMT error messages for attempts to set the persistent session delay interval (continued)

Message	RESP/RESP2
BACK LEVEL VTAM	RESP=INVREQ, RESP2=10
ACB CLOSED	RESP=INVREQ, RESP2=12
RECOVERY ERROR	All other cases of INVREQ

CEMT SET WEB

Change Web garbage collection settings, or Web 3270 terminal timeout settings.

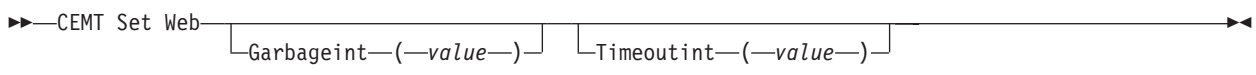
Input

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET WEB` (the minimum abbreviation is `CEMT S WE`). You get a display that lists the current status, similar to that obtained by CEMT INQUIRE WEB. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET WEB` (`CEMT S WE`) followed by one or more attribute settings that you wish to change. For example, `cemt s we g` specifies that you want to change the GARBAGEINT option.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET WEB



Displayed fields

GARBAGEINT
is the interval at which the Web garbage collection task runs to clean up Web 3270 state data for which the terminal timeout interval has expired. The interval is measured in minutes.

TIMEOUTINT
is the period of time after which inactive Web 3270 sessions are eligible for garbage collection. The period is measured in minutes.

CEMT SET WEBSERVICE

Function

Set the validation status of a WEBSERVICE.

Description

If the optional attributes are omitted, the command has no effect.

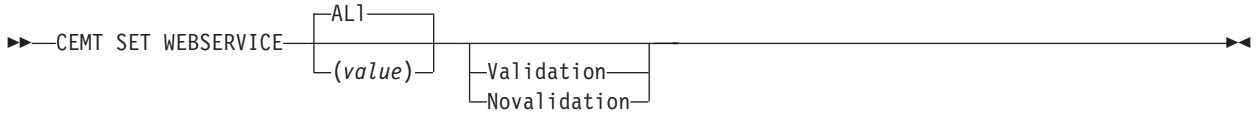
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET WEBSERVICE` (the minimum abbreviation is `CEMT S W`) followed by one or more WEBSERVICE identifiers or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE WEBSERVICE. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET WEBSERVICE` (`CEMT S W`) followed by one or more WEBSERVICE identifiers or ALL, followed in turn by one or more attribute settings that you wish to change. For example, `cemt s we al v` resets the validation status for all WEBSERVICEs.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

SET WEBSERVICE



Conditions: NOTAUTH, NOTFND

Options

ALl
Any changes you request are made to all resources of the specified type that you are authorized to access.

(*value***)**
specifies the 1–8 character name of the WEBSERVICE.

Validationst
Specifies whether full validation of SOAP messages is required for this WEBSERVICE:

Validation
Full validation is required.

Novalidation
Full validation is not required.

The WEBSERVICE must be INSERVICE before you can turn validation on and off.

CEMT SET WORKREQUEST

Function

Purge or Forcepurge a specific work request task.

Description

SET WORKREQUEST enables you to terminate specific work requests. You cannot specify a list of identifiers, nor can you use the symbols * and + to specify a family of work requests.

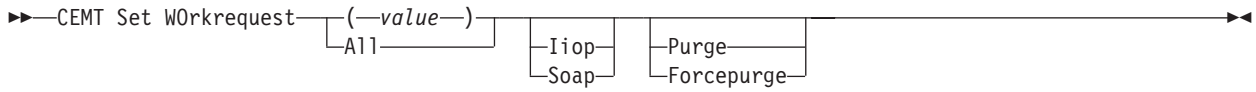
Syntax

Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- Type `CEMT SET WORKREQUEST` (the minimum abbreviation is `CEMT S WO`) with (*value*) or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE WORKREQUEST. You can then tab to the Purgetype field and enter or overtype it with the required value (PUrge or FOrcepurge).
- Type `CEMT SET WORKREQUEST` (`CEMT S WO`) with (*value*) or ALL, followed by the attribute setting that you wish to change. For example, `cemt s wo(value) pu` will purge the specified work request.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

CEMT SET WORKREQUEST



Options

(*value***)**
An 8-character string to identify the work request. The valid characters for this token are the 16 hexadecimal digits, 0–9 and 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.

All
Any action you request is performed for all work requests of the specified type that you are authorized to access.

Forcepurge
The work request is terminated immediately. System integrity is not guaranteed. In some extreme cases, for example if a work request is forcepurged during backout processing, CICS terminates abnormally. If you want to terminate a work request but do not want to terminate CICS, you should use PURGE instead of FORCEPURGE.

Iiop
Specifies that any action you request is limited to IIOP work requests.

Soap
Specifies that any action you request is limited to Web service work requests.

Purge
The work request is terminated, but termination occurs only when system and data integrity can be maintained.

Chapter 16. CEOT - terminal status

The CEOT command allows you to inquire about the status of your own terminal, and to change some of its values. You cannot use CEOT to inquire about or change the values of any other terminal.

A number of attributes on CSD resource definitions, particularly those that refer to Java applications, are specified using a mixed-case character set. The CEOT upper case translate status options (see “CEOT in set mode” on page 468) enable you to switch between the uppercase translation options to suit a specific requirement. For example, you might want to suppress CICS uppercase translation before using the CEDA transaction to define resources with attributes that require mixed-case input, such as HFSFILE, SHELF, CERTIFICATE, or DESCRIPTION.

You can start CEOT from 3270 family display devices, or from equivalent terminals, or from the operating system console.

Press the Clear key to clear the screen. There are two ways of using this transaction: in inquiry mode and in set mode.

Using CEOT in inquiry mode

Type `CEOT`. You get the following display:

```
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Ter(TC37) Trans(CEOT) Pri(000) Pag Ins Ati Tti
    Net(IYCQTC37) Acq Tra

 CEOT SYNTAX:
  < PAgeable | AUtopageable >
  < ATi | NOAti >
  < TTi | NOTti >
  < Uctran | NOUctran | TRANIdonly >









                                          SYSID=HT61 APPLID=CICSHT61
  RESPONSE: NORMAL                   TIME:  13.47.07  DATE: 02.01.01
PF 1 HELP       3 END       5 VAR           9 MSG
```

Figure 177. CEOT transaction: initial screen

The top half of the screen shows the current status of your terminal. The bottom half of the screen shows the syntax of the CEOT command.

Note: NET, ACQ, and CRE are not displayed for non-VTAM terminals.

The status items are displayed in the following sequence.

Note: If an item has a “negative” value—for example, NOATI—that value will be replaced by a blank space.

© Copyright IBM Corp. 1982, 2011

Acq
CICS is in session with the logical unit represented by this terminal. You cannot overtype this value.

Aut
See 'AUTOPAGEABLE' in the Command Options below.

The paging status can be modified by overtyping.

ATi
See 'ATI' in the Command Options below.

Cre
This terminal can be acquired automatically by ATI transactions. You cannot overtype this value.

Ins
Your terminal is INSERVICE. You cannot overtype this value.

Net (*netname***)**
displays the name by which this terminal is known to VTAM.

Pag
See 'PAGEABLE' in the Command Options below.

The paging status can be modified by overtyping.

Pri (*nnn***)**
displays the priority of your terminal relative to other terminals, and is a number in the range 0–255.

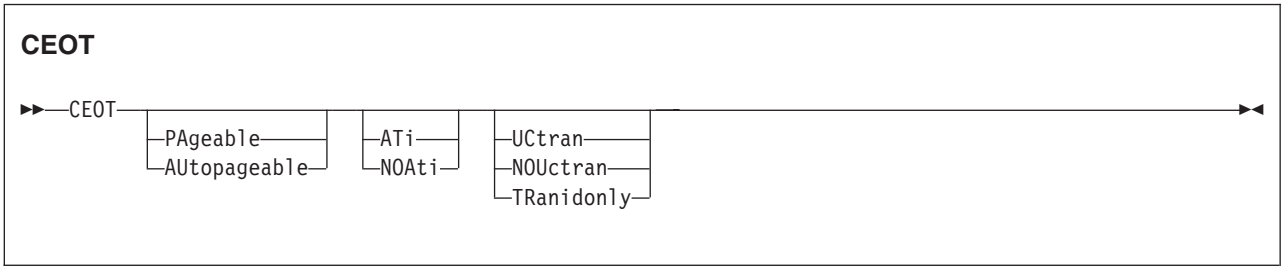
Ter (*xxxx***)**
displays the 4-character identifier of your terminal.

Tra (*ceot***)**
displays the transaction that is running on your terminal. This is always CEOT.

Tti
The terminal is available for use by transactions initiated from this terminal. You cannot change the TTI setting by overtyping it with blanks.

CEOT in set mode

Type `CEOT` followed by one or more of the following attributes:



ATi
Your terminal is available for use 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.

AUtopageable
Pages after the first in a series are written to the terminal automatically. This option should never be used with a display device.

NOAti
Your terminal is not available for use 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.

NOTti
This terminal cannot be used by transactions that are initiated from this terminal. Setting NOTTI is valid only for sequential devices.

NOUCTRAN
The uppercase translation status of your terminal is set to OFF for the current session.

PAgeable
Pages after the first in a series are written to the terminal on request from the operator.

TRANIDONLY
The uppercase translation status of your terminal is set to translate only transaction identifiers entered at the terminal for the current session.

TTi
This terminal can be used by transactions.

UCTRAN
The uppercase translation status of your terminal is set to ON for the current session.

Typing ? at the beginning of the first line gives a syntax prompt. Resetting the values takes effect immediately.

If you do not want to view the current status of your terminal (as displayed in the top half of the CEOT screen), you can change the paging status and the ATI status by using the CEOT syntax displayed in the bottom half of the screen. So, for example, if you simply want to make your terminal PAGEABLE, type `CEOT PAGEABLE` on the command line of your display and press the Enter key.

Note:

1. You can use CEOT with the routing transaction (CRTE).
   - A user can inquire only about his or her own terminal, and cannot change any of the attributes.
   - The attributes returned are those of the remote terminal TCTTE in the application-owning region(AOR).
   - You must cancel the CRTE transaction by typing CANCEL on the command line.
2. Using CEOT in the TOR:
   - When a terminal is defined as unshippable, and CEOT is used to change the status of the terminal, the changes are not reflected in a remote terminal built in an AOR as a result of a transaction routing request. The status of the remote terminal is determined by the remote definition of the terminal in the AOR.
   - Even when a terminal is defined as shippable, a status change made in the TOR, using CEOT, is not reflected in an existing remote terminal definition in the AOR. A remote terminal in the AOR is created from a shippable definition the first time it is required in the AOR for running a

program. The status of the remote terminal returned is its current status in the TOR. Any later updates are not passed on to the remote terminal.

3. If you make an invalid request, CICS lists the keyword options and waits for a valid keyword entry.

Chapter 17. CESF - sign off

Use the CESF transaction to sign off from the CICS system.

This transaction:

- Writes the signoff message to your terminal
- Logs the message on the transient data destination CSCS
- Logs any error message on the transient data destination CSMT.

If you are using CESF to disconnect a terminal and security is active, a message will still be sent to the log if the signoff fails. However, this will not stop the terminal being disconnected if it is a preset terminal or if the terminal is not signed on.

You cannot use CESF on:

- Fixed-format terminals (such as the 3608, the 3614, and the 3653)
- Numeric-only terminals.

For these devices, you must code your own transaction using the EXEC CICS SIGNOFF command. For programming information about this command, see the *CICS Application Programming Reference*.

The syntax of the CESF transaction is as follows:

CESF

If you simply type CESF by itself, you are signed off the CICS system. Your terminal can still start transactions without you having to request them, and messages can be sent to it. You can sign on again at that terminal at any time.

The effects of typing CESF with the GOODNIGHT value or the LOGOFF value depend on the access method in use, as shown in Table 9:

Table 9. Effect of CESF LOGOFF and GOODNIGHT with different access methods

	VTAM DISCREQ=YES	**Otherwise**
CESF	TTI	TTI
CESF LOGOFF	Disc, TTI	TTI
CESF GOODNIGHT	Disc, TTI	NOTTI

where:

Disc The terminal is disconnected.

TTI Terminal transaction initiation is allowed.

© Copyright IBM Corp. 1982, 2011

NOTTI Terminal transaction initiation is not allowed, and your terminal is placed in RECEIVE status so that, the next time a connection is made, it can receive messages.

DISCREQ=YES is an operand of the TYPETERM definition associated with the terminal, and means that CICS is to honor a disconnect request. If DISCREQ=NO, a disconnect request (for example, a CESF LOGOFF request) is not honored.

If a CESF GOODNIGHT is given when there are outstanding automatic transaction initiation (ATI) requests, the ATI requests are actioned and the session is reacquired irrespective of the RDO options.

Chapter 18. CESN - sign on

By default, intending users of CICS see the CICS “good-morning” panel displayed when they logon to CICS as a VTAM application (see Figure 178).

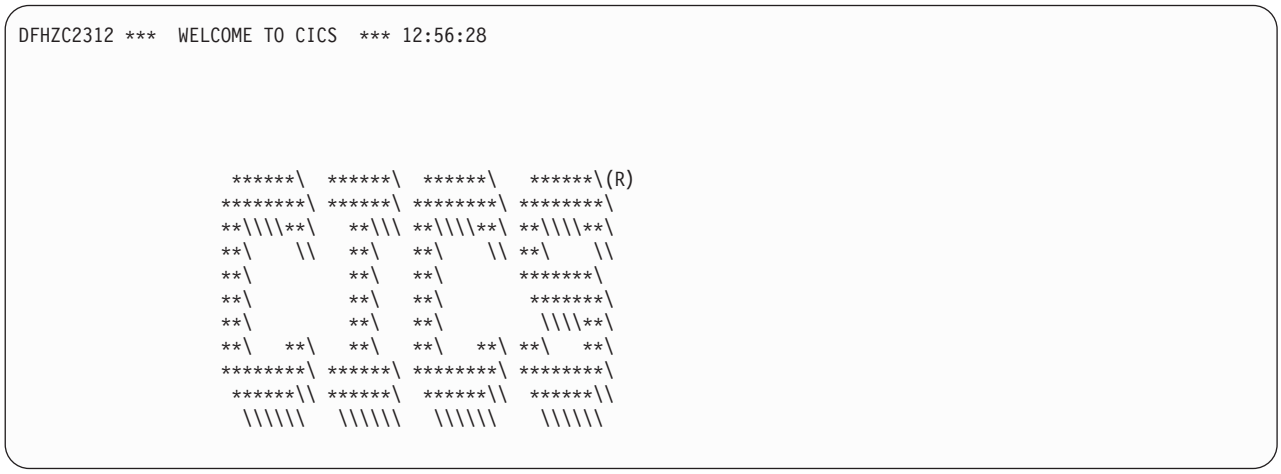


Figure 178. The CICS good-morning panel

This panel is displayed by the CICS-supplied transaction, CSGM. Instead of using CSGM, you can specify your own transaction panel on the GMTRAN system initialization parameter.

Users may either press the CLEAR key and type the CICS sign-on transaction id, CESN or, if CESN is specified as the good-morning transaction, CICS displays the sign-on panel shown in Figure 179 on page 474.

The user ID specified for this transaction is used in error messages and so on, to identify the user.

Note:

1. If you sign on at a terminal that is already signed on, CESN signs off the previous operator. This does not happen, however, unless you complete the CESN panel before pressing ENTER. For further information, see “Signing off with CESN” on page 476.
2. For numeric-only devices or for the IBM 2741 Communication Terminal, use the signon procedure described in “Non-3270 family terminal” on page 475.
3. Userids and passwords specified in the CESN transaction are translated to uppercase, ignoring the setting of the UCTRAN attribute. However, if support for mixed case passwords is active in the external security manager, CICS saves the value of the UCTRAN option and temporarily sets it to UCTRAN(NO) while the user is prompted to enter a userid and password. CESN then uppercases the userid but leaves the password in the case entered by the user. When the CESN transaction completes, the UCTRAN attribute is restored to its original value. If the user disconnects from the terminal before the CESN transaction has completed, the terminal might be left with the UCTRAN(NO) attribute set.
4. Autoinstall of BMS maps is not carried out within the CESN transaction. When CESN tries to send out any BMS map that is not installed, CESN will send out the English BMS map.

© Copyright IBM Corp. 1982, 2011

5. This dialog will not prompt you for a new password unless your old password has expired. If you wish to change your password before it has expired, you must use another method, such as signing on at a CICS terminal rather than a console.

IBM 3270 family display devices

Clear the screen, and type CESN on the command line of the screen. Press ENTER. You get the following display:

```
         Sign on for CICS Transaction Server                APPLID=CICSA100

. . . . . . This is where the good morning message appears. . . . . .
. . . . . . It can be up to four lines in depth to contain the. . . . .
. . . . . . maximum message length of 246 characters (that is, three. .
. . . . . . full lines and six characters on the fourth line) . . . . .



Type your userid and password, then press ENTER:

         Userid . . . . ________    Groupid . . . ________

         Password . . . ________

         Language . . . ___

     New Password . . . ________




DFHCE3520 Please type your userid.
F3=Exit
```

Figure 179. CESN transaction: initial screen

The panel for 40 x 12 size screens is shown below. There is no space for the release number or the good morning message.

```
         Sign on for CICS Transaction Server                APPLID=CICSA100

            Userid   . . . . ________
            Groupid  . . . . ________
            Password . . . . ________
            Language . . . . ___
            New Password . . ________







DFHCE3520 Please type your userid.
F3=Exit
```

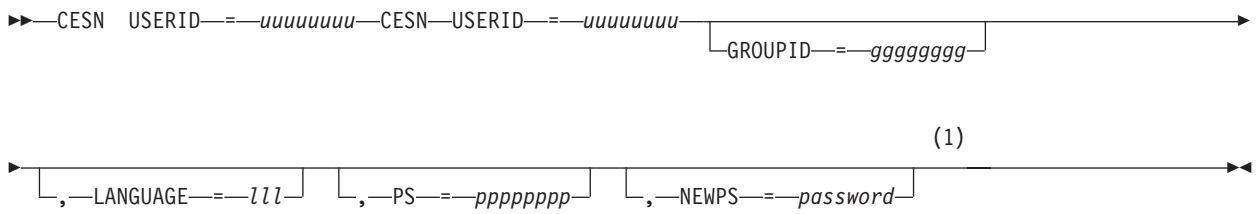
Figure 180. 40 x 12 screen

Enter your user ID. This may be up to 8 characters in length.

Non-3270 family terminal

A user can also sign on by providing all the user information along with the transaction id, as follows:
This style of input can be used on video display terminals, but is especially useful

CESN



Notes:

1 Do not use blanks between parameters in the comma-separated list.

for sequential terminals and devices not belonging to the IBM 3270 family of terminals.

Command options

GROUPID=gggggggg
is your group userid (up to 8 characters). If this parameter is specified, it may be propagated to remote systems along with the userid (refer to the section on 'Implementing LU6.2 Security / Information about remote users' or 'Implementing MRO security / Information about remote users' in the*CICS RACF Security Guide*.

LANGUAGE=lll
enables you to choose a national language (from those that are available at your installation) for terminal directed messages. “lll”, the national language identifier, is a standard, three-character IBM code. If the language identifier is omitted, CICS examines the following in this order:

1. Language specified for the user in the signon table
2. Language specified in the installed terminal definition for the terminal at which the signon occurs
3. Language specified on the NATLANG system initialization parameter.

If you enter a code other than the range of IBM codes, you get the following message:

```
DFHCE3537 Language is invalid. Please retype.
```

NEWPS=password
“password” is your new password (up to 8 characters). A ‘password’ value that consist entirely of space characters will not be interpreted as a password update request. The user will not be prompted for verification.

PS=pppppppp
is your password (up to 8 characters). If the password is required but omitted, then RACF will report a security violation via message ICH408I before CICS prompts for the password.

USERID=uuuuuuuu
is your user ID (up to 8 characters).

Signing off with CESN

When a user starts the CESN transaction, CICS verifies userids and password by calling RACF. If the terminal user sign-on is valid, the CICS user domain keeps track of the signed-on user. Thereafter, CICS uses the information about the user when calling RACF to make authorization checks. If a user was signed on before the call to RACF, this signed-on user is signed off by CICS whether the returns from RACF are valid or invalid.

If users enter the CESN transaction when they are already signed on, the transaction may or may not sign them off. For example, a user is not signed off by entering CESN, and then pressing F3 when the signon panel is displayed.

In CICS TS for z/OS, Version 3.2, CESN signs off any signed-on user only when a new signon attempt is made. A signed-on user is signed off when a userid is entered on the signon panel and the ENTER key is pressed. Alternatively, the signed-on user is signed off if the CESN transaction identifier is entered with operands (for example USERID=userid), and the ENTER key is pressed.

Unsuccessful sign-on and sign-off errors cause DFHCEnnnn messages to be issued.

RACF OIDCARD users can use CESN to sign on if the card reader supports the DFHOPID attention identifier (AID). If it does not, you will need to write your own sign-on program using the EXEC CICS SIGNON command. For programming information see the *CICS Application Programming Reference*.

Chapter 19. CEST—supervisory terminal

Use CEST to invoke a subset of the master terminal (CEMT) transaction. The CEST INQ|SET commands allows you to inquire about and alter some of the values of lines, netnames, tasks, and terminals.

CEST can be invoked from 3270 family display devices, or from equivalent terminals, or from the operating system console.

All of the CEST commands require the SUPRID keyword to be specified. SUPRID identifies the supervisor invoking the transaction, and its value is the 1-or 2-character suffix of a terminal list table (TLT). For information about the SUPRID keyword, see the *CICS Resource Definition Guide*.

For example, CEST INQ TERMINAL SUPRID(AL) displays the status of all the terminals in the DFHTLTA1 table. Values can be changed by overtyping the display or by using CEST SET commands. Apart from the requirement that SUPRID has to be specified, the syntax of the commands and the meaning of the values is exactly as described in CEMT INQUIRE|SET LINE, NETNAME, TASK, and TERMINAL.

© Copyright IBM Corp. 1982, 2011

Chapter 20. CETR - trace control

Use CETR to control CICS tracing activity.

Start the transaction by typing CETR on the command line of your display, as follows:

```
CETR
```

Press the ENTER key.

You get the following display, with the values of your own system in each of the fields.

```
CETR                   CICS Trace Control Facility

Type in your choices.

Item                               Choice       Possible choices

Internal Trace Status       ===>  STOPPED      STArted, STOpped
Internal Trace Table Size   ===>  16      K    16K - 1048576K

Auxiliary Trace Status      ===>  PAUSED       STArted, STOpped, Paused
Auxiliary Trace Dataset     ===>  B            A, B
Auxiliary Switch Status     ===>  ALL          NO, NExt, All

GTF Trace Status            ===>  STARTED      STArted, STOpped

Master System Trace Flag    ===>  OFF          ON, OFf
Master User Trace Flag      ===>  OFF          ON, OFf

When finished, press ENTER.




PF1=Help    3=Quit    4=Components    5=Ter/Trn    6=JVM    9=Error List
```

Figure 181. CETR transaction: initial screen

Trace field descriptions

In this example:

- Internal tracing status is STOPPED, and so regular tracing is not directed explicitly to the internal trace table. However, note that the internal trace table is used as a buffer for the other trace destinations, so it always contains the most recent trace entry if at least one trace destination is STARTED.

  The internal trace table is also used as a destination for exception trace entries, which are made whenever CICS detects an exception condition. If such a condition is detected when the options shown in this example are in effect, you would be able to find the exception trace entry in the internal trace table as well as in the GTF trace data set.
- The internal trace table size is 16KB, which is the minimum size it can be. If internal trace were STARTED, the trace table would wrap when it became full.
- The current auxiliary trace data set is B, meaning that trace entries are written to DFHBUXT if auxiliary tracing is started. As its status is shown to be PAUSED, no tracing is done to that destination. The auxiliary switch status is ALL, so a switch would be made to the other auxiliary trace data set whenever one became full.

© Copyright IBM Corp. 1982, 2011

- The GTF trace status is shown to be STARTED, which means that CICS trace entries are written to the GTF trace data set defined to MVS. Be aware that no error condition is reported if the CICS GTF status is started but GTF tracing has not been started under MVS. If this happens, the trace entries are not written.
- The master system trace flag is OFF. This means that no standard tracing is done at all, even though standard tracing might be specified for some tasks. However, special task tracing is not affected. The master system trace flag only determines whether standard task tracing is to be done.

Any of the input fields can be overtyped with the new values that you require. When you press ENTER, CETR issues the necessary commands to set the new values. If there are any errors, PF9 can be pressed to display the error messages. If there is only one short error message, it appears near the bottom of this display.

The following logic is used to ensure that trace entries are written to the required destinations:

1. The trace entry is built in the internal trace table.
2. If auxilliary tracing status is STARTED, the trace data is copied to the current auxilliary trace data set.
3. If GTF tracing status is STARTED **and** GTF tracing is started under MVS with the TRACE=USR option, the trace data is copied to the GTF trace data set.

The following table shows the relationships between the auxiliary trace status, trace flag, and trace data set.

Table 10. The meanings of auxiliary trace status values

Auxiliary tracing status	Auxiliary trace flag	Auxiliary trace data set
Started	On	Open
Paused	Off	Open
Stopped	Off	Closed

Three other main functions of CETR are:

1. Setting the component trace options (press PF4)
2. Setting the transaction and terminal trace options (press PF5).
3. Setting the JVM trace options (press PF6).

Use caution when setting TRTABSZ to a very high value because there must be enough MVS page storage to satisfy both the request and DSA sizes. Use the system command `DISPLAY ASM MVS` to display current information about status and utilization of all MVS page data sets.

For information about the use of the various CETR options as an aid to problem determination, see the *CICS Problem Determination Guide*

Using component trace options

The “Component Trace Options” display is produced by pressing PF4 on the main CETR display. You get the following display, with the values of your own system in each of the fields.

```
CETR                  Component Trace Options                  CI41 IYAHZCCV

Over-type where required and press ENTER.                      PAGE 1 OF 4
Component Standard                         Special
--------  -------------------------------  -------------------------------
  AP      1                                1-2
  BA      1                                1-2
  BM      1                                1
  BR      1                                1-2
  CP      1                                1-2
  DC      1                                1
  DD      1                                1
  DH      1                                1-2
  DM      1                                1-2
  DS      1                                1-2
  DU      1                                1-2
  EI      1                                1
  EJ      1                                1-2
  EM      1                                1-2
  FC      1                                1-2
  GC      1                                1-2
  IC      1                                1

PF:  1=Help   3=Quit   7=Back  8=Forward  9=Messages   ENTER=Change
```

Figure 182. CETR transaction: component trace options screen

PF7 and PF8 can be used to scroll backward and forward through the list of components. PF9 gives the error message list.

The trace levels for a particular component are represented by two values. One gives the active level of tracing for standard tracing; the other gives the level for special tracing. From the list of components, you can set the active level of tracing for standard and/or special tracing for each individual component. See the *CICS Problem Determination Guide* for an explanation of standard and special tracing, and of tracing levels.

The values are in the range 1 through 32 for most components. In practice, you will normally want to use an active level of ‘1’ for any component, and you do this by overtyping (if necessary) the displayed value under the relevant heading for the required component.

In certain circumstances, if you have referred a problem to the IBM System Support Center staff, they may ask you to get a level ‘2’ trace. You will need to specify both level 1 and 2 to get all of the trace points that are needed.

You can specify a range of tracing levels—normally levels ‘1’ and ‘2’. You do this by overtyping the displayed value with ‘1-2’ or ‘1,2’.

Alternatively, you can type ‘ALL’ in upper or lower case under the relevant heading. Similarly, you can type ‘OFF’ to turn off the tracing for a particular component under either the ‘special’ or ‘standard’ heading.

The equal sign “=” can be used, when changing several fields, to indicate that a particular field should be changed to the same value as the one above.

The complete list of components is:

Component	Title
AP	Application domain
BA	CICS business application manager for BTS

Component	Title
BF*	Built-in function support
BM*	BMS
BR*	3270 Bridge
CP*	Common Programming Interface
DC*	Dump macro-compatibility
DD	Directory manager domain
DH	Document handling domain
DI*	Data interchange
DM	Domain manager domain
DP	Debugging Profiles domain
DS	Dispatcher domain
DU	Dump domain
EI*	Exec interface
EJ	Enterprise Java domain
EM	Event manager domain for BTS
FC*	File control
GC	Global catalog domain
IC*	Interval control
IE	ECI over TCP/IP domain
II	IIOP domain
IS*	Intersystems communication and MRO
KC*	Task control macro-compatibility
KE	Kernel
LC	Local catalog domain
LD	Loader domain
LG	Log manager domain
LM	Lock manager domain
ME	Message domain
MN	Monitoring domain
NQ	Enqueue manager domain
OT	Object transaction services domain
PA	Parameter manager domain
PC*	Program control
PG	Program manager domain
PI	Pipeline manager domain
PT	Partner Management
RI*	Resource manager interface (RMI)
RM	Recovery manager domain
RX	Recoverable EXCI domain
RZ	Request streams domain
SC*	Storage control macro-compatibility
SH	Scheduler services domain for BTS
SJ	JVM domain (but use JVM trace options screens to set JVM trace levels: see note below)
SM	Storage manager domain
SO	Sockets domain
ST	Statistics domain
SZ*	Front End Programming Interface
TC*	Terminal control
TD*	Transient data
TI	Timer domain
TR	Trace domain
TS	Temporary storage domain
UE*	User exit control

Component	Title
US	User domain
WB	Web interface
XM	Transaction manager
XS	Security manager domain

Note:

1. If you use CETR to turn on tracing for a particular component, CICS does not save modifications to the component trace flags across any kind of CICS restart.
2. 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.
3. The SJ domain, which controls JVM tracing, is a special case. As well as using the normal trace levels, the SJ domain uses trace levels 29–32. 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. You cannot use the component trace options screen to control JVM tracing; use the JVM trace options screens instead, as described in “Specifying JVM trace options in CETR” on page 485.
4. The SZ component can be used only if the Front End Programming Interface feature (FEPI) is installed. For information, see the *CICS Front End Programming Interface User's Guide*.
5. For the DS domain function CHANGE_MODE, a trace entry is generated if DS level 2 or 3 tracing is active.

For information about the use of trace for problem determination, and for information about the format of trace entries, see the *CICS Problem Determination Guide*.

Using Transaction and terminal-related trace options

The “Transaction and Terminal Trace” display is produced by pressing PF5 on the main CETR display. It can be used to set STANDARD, SPECIAL, or SUPPRESSED tracing for particular CICS tasks. This is done by setting the appropriate value for the transaction ID used to start the task, or for the terminal from which the task is to be run. Note that SUPPRESSED cannot be set for terminals, only for transaction IDs. The required terminal can also be specified by using its netname.

Setting SPECIAL tracing for a particular transaction means that the SPECIAL flags for each component (as defined on the “Component Trace Options” display) is used to decide whether a particular trace entry is to be made at any given point in the process of the task.

The display layout is as follows:

```
CETR                  Transaction and Terminal Trace           CI41 IYAHZCCV

Type in your choices.

Item                               Choice       Possible choices

Transaction ID              ===>               Any valid 4 character ID
Transaction Status          ===>               STandard, SPecial, SUppressed

Terminal ID                 ===>               Any valid Terminal ID
Netname                     ===>               Any valid Netname
Terminal Status             ===>               STandard, SPecial
Terminal VTAM Exit Trace    ===>               ON, OFf
Terminal ZCP Trace          ===>               ON, OFf

VTAM Exit override          ===>  NONE         All, System, None


When finished, press ENTER.




PF1=Help      3=Quit      6=Cancel Exits      9=Error List
```

Figure 183. CETR transaction: transaction and terminal trace screen

A common way of using this display is to set the terminal trace status for a particular terminal to SPECIAL, to have all of the STANDARD component trace flags as level-1 only, and the SPECIAL component trace flags as level-1 and level-2. In this way, all tasks run from the SPECIAL terminal have level-1 and level-2 tracing. All other tasks have level-1 tracing only.

For example, you can use SPECIAL tracing to trace a single transaction. You can do this by setting the “master system trace flag” OFF on the main CETR display. This turns STANDARD tracing OFF. (STANDARD tracing can also be turned off by coding SYSTR=OFF at system initialization.) You can then press PF4 for the “components trace options” screen, shown in Figure 182 on page 481, where you select the levels of tracing for the required components under the heading “Special”. Finally, on the “transaction and terminal trace” screen, shown in Figure 183, specify the transaction ID and a transaction status of SPECIAL.

This display can also be used to enable VTAM exit tracing and ZCP trace for specified terminal IDs or netnames. To set VTAM exit tracing in an AOR, enter the applid of the TOR as Netname and enter ON for the Terminal VTAM Exit Trace.

If dynamic LU alias is in operation for the CICS region, and you want to use VTAM exit tracing to trace the bind flows for an autoinstalled terminal, the NETNAME you specify should be the real network name. However, if you want to trace when the LU alias is known, specify the LUALIAS name. If the real network name is used, and there is more than one network using that name, the VTAM exit tracing is activated for each occurrence of the network name.

There are also fields that allow enabling of all nonterminal VTAM exits, and PF6 can be used to cancel all exit tracing.

Specifying JVM trace options in CETR

The “JVM Trace Options” display is produced by pressing PF6 on the main CETR display. You can use these screens to display and update trace settings for JVMs. Although the JVM trace options are part of the SJ component, they are controlled using the JVM trace options screens, rather than the component trace options screen. Trace levels 29–32 for the SJ component correspond to JVM Level 0, Level 1, Level 2, and User trace respectively.

You can use the “JVM Trace Options” display to specify the JVM trace options, using the "free-form" 240–character field, and to specify trace settings for JVMs using the Standard and Special flags. You can then use the Transaction and Terminal Trace screen to switch on these flags for particular transactions. JVM trace can produce a large amount of output, so you should normally activate JVM trace for special transactions, rather than turning it on globally for all transactions.

There are two JVM trace options screens, one for JVM Level 0 trace and JVM Level 1 trace, and one for JVM Level 2 trace and JVM User trace. The screens display the trace flags and the trace options strings associated with each flag. Press PF8 to move to the second JVM trace options screen, and PF7 to move from there back to the first screen. Set any of the trace options to ‘*’ to reset the option to its default settings, or press PF6 to reset all of the JVM trace options to their default settings. Press PF3 to return to the CETR main screen, and press Enter on any screen to pass the changes to CICS.

The display layout is as follows:

```
 CETR                          JVM Trace Options                CJB2 IYK2Z1V2

 Type in your choices.                                          PAGE 1 OF 2

 Item                             Standard            Special

 JVM Level 0 Trace           ===>  OFF  ON, OFf  ===>  OFF  ON, OFf
 Option String:
LEVEL0



 JVM Level 1 Trace           ===>  OFF  ON, OFf  ===>  OFF  ON, OFf
 Option String:
LEVEL1



 When finished, press ENTER.



 PF1=Help    3=Quit    6=Reset             8=Forward    9=Error List
```

Figure 184. CETR transaction: JVM trace options screen 1 of 2

```
CETR                          JVM Trace Options                 CJB2 IYK2Z1V2

 Type in your choices.                                          PAGE 2 OF 2

 Item                              Standard            Special

 JVM Level 2 Trace           ===>  OFF  ON, OFf  ===>  OFF  ON, OFf
 Option String:
LEVEL2



 JVM User Trace              ===>  OFF  ON, OFf  ===>  OFF  ON, OFf
 Option String:
NONE



 When finished, press ENTER.




 PF1=Help    3=Quit    6=Reset    7=Back                9=Error List
```

Figure 185. CETR transaction: JVM trace options screen 2 of 2

The *CICS Problem Determination Guide* has information about the JVM trace options that you can set using the JVM Level 0 trace, JVM Level 1 trace, JVM Level 2 trace, and JVM User trace levels. The default JVM trace options that are provided in CICS use the JVM trace point level specifications. You can add further parameters to the basic level specifications for JVM Level 0 trace, JVM Level 1 trace, and JVM Level 2 trace, if you want to include or exclude particular components or trace point types at the selected trace levels. If you want to create more complex specifications for JVM tracing, use the JVMUSERTRACE option to create a trace option string that includes the parameters of your choice.

When you activate JVM trace, the results appear as CICS trace points in the SJ (JVM) domain. Each JVM trace point that is generated appears as an instance of a CICS trace point:

- SJ 4D02 is the trace point used for formatted JVM trace information.
- SJ 4D01 is used for any JVM trace points that cannot be formatted by CICS. If you see this trace point often, check that the trace format file supplied with the IBM SDK for z/OS, Java 2 Technology Edition is present in the /lib/ subdirectory of your SDK installation. For Version 1.4.2 of the SDK, the file is called TraceFormat.dat, and for Version 5 it is called J9TraceFormat.dat. CICS requires this file to format the JVM trace points.

If the JVM trace facility fails, CICS issues the trace point SJ 4D00.

There is further information about JVM trace and about problem determination for JVMs in the *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition Diagnostics Guide*, which is available to download from www.ibm.com/developerworks/java/jdk/diagnosis/.

Using CETR HELP panels

If you press the help key (PF1), the following panels are displayed:

```
CETR                Help: CICS Trace control facility


 FUNCTION OF THE MAIN PANEL.

  This panel displays the current state of CICS Internal, Auxiliary
  and GTF trace and the settings of the master system and user trace
  flags.  The current values are obtained from the system on initial
  entry and subsequently every time ENTER or a PF key is pressed.
  The values can be altered by over-typing and pressing ENTER.

  The APPLID of the CICS system being queried is displayed in the top
  right corner of the screen.



 Further help information follows:

         1. Data entry and alteration.
         2. Description of fields.
         3. Function key use on the main panel.


PF 3=End  8=Next  ENTER=End
```

Figure 186. CETR transaction: HELP panel (1)

```
CETR                Help: CICS Trace control facility


 (1) DATA ENTRY AND ALTERATION.

  This is achieved by simply over-typing the existing data, and then
  pressing ENTER.

  Abbreviations may be used, provided they are not ambiguous. The minimum
  possible abbreviations are listed in capitals to the right hand side of
  the input fields. For example STO may be entered for STOPPED, but ST
  would not be allowed as it could mean STARTED or STOPPED.

  Erased or blank fields will redisplay the current level settings as
  known to CICS. These will be the levels current when ENTER was last
  pressed.  They may have been changed from the previously displayed
  values by other activity in the system.

  All input will be validated when ENTER is pressed, and no internal
  settings will be made until all errors have been removed.



PF 3=End  7=Back  8=Next  ENTER=End
```

Figure 187. CETR transaction: HELP panel (2)

```
CETR               Help: CICS Trace control facility

 (2) DESCRIPTION OF FIELDS.

 The values that can be set from the main panel are:-

  Internal Trace Status     -  Whether tracing to the main storage trace
                               table is required.
  Internal Trace Table Size -  The size of the main storage trace table.
  Auxiliary Trace Status    -  Whether tracing to the current auxiliary trace
                               data set is required.
  Auxiliary Trace Dataset   -  The current data set. A = DFHAUXT, B = DFHBUXT.
  Auxiliary Switch Status   -  Whether a switch to the alternate extent is
                               required at next or all end-of-extents.
  GTF Trace Status          -  Whether CICS is writing its trace data to GTF.
                               Note that GTF must have been started to MVS.
  Master System Trace Flag  -  This flag must be on if CICS system-requested
                               entries are to be written for STANDARD tracing
                               tasks.
  Master User Trace Flag    -  This flag must be on if user-requested trace
                               entries are to be produced.


PF 3=End  7=Back  8=Next  ENTER=End
```

Figure 188. CETR transaction: HELP panel (3)

```
CETR               Help: CICS Trace control facility


 (3) FUNCTION KEY USE ON THE MAIN PANEL.

         Key     Name        Function

         ENTER   Change      Execute any changes.
         PF1     Help        Display this help information.
         PF3     Quit        Terminate the transaction.
         PF4     Components  Display or set STANDARD or SPECIAL trace levels
                             for individual CICS components.
         PF5     Ter/Trn     Query or set current trace status for terminal
                             or transaction.
         PF6     JVM         Query or set current JVM trace options
         PF9     Messages    Display any messages created, these can be
                             warning (informative) messages, or errors.









PF 3=End  7=Back  ENTER=End
```

Figure 189. CETR transaction: HELP panel (4)

HELP panels for components

If you press the help key (PF1) while viewing the component trace options screen, the following panels are displayed:

```
CETR            Help: CICS Component Trace

  FUNCTION OF THE COMPONENT TRACE PANEL

   The component trace panel is used to inquire and set the STANDARD and
   SPECIAL trace levels for the individual components of the CICS system.

   Every component of CICS has a number of trace points.  Associated with
   each is a trace level.  These levels are numbers from one to thirty-two.
   For the trace entry associated with the trace point to be made, the
   associated level must be active (except for exception traces).

   Each CICS task is either a STANDARD tracing task or a SPECIAL tracing
   task.  This distinction is made at task attach time when the STANDARD or
   SPECIAL attributes of the terminal id and transaction id are examined.
   These can be set on the Terminal and Transaction Trace panel of CETR.

  Further help information follows:
        1. Data entry and alteration 1
        2. Data entry and alteration 2
        3. Meanings of Component abbreviations.
        4. Function key use on the component trace panels.

PF 3=End  8=Next  ENTER=End
```

Figure 190. CETR transaction: Component trace HELP panel (1)

```
CETR            Help: CICS Component Trace

 (1) DATA ENTRY AND ALTERATION 1

 From this panel detailed tracing levels can be set for each component.
 Each level is specified by a number between one and three, the
 presence of the number indicates that the level is set, the omission
 of a number indicates that the level is not set.

 Multiple consecutive levels are displayed as xx-yy, where xx is the
 lower of the consecutive values, and yy is the higher.  For example,
 if 1-3 was displayed, then trace levels 1,2 and 3 would all be set.

 There is more than one page of components.  Function keys PF7 and PF8
 can be used in the usual way to switch pages. The page may only be
 switched when the displayed page contains no errors.

 Changes to the level settings are automatically validated when the
 current page is changed, but these levels are only set within CICS when
 ENTER is pressed. Until this time it is possible to quit from the
 program (PF3) without altering internal values.


PF 3=End  7=Back  8=Next  ENTER=End
```

Figure 191. CETR transaction: Component trace HELP panel (2)

```
CETR            Help: CICS Component Trace

 (2) DATA ENTRY AND ALTERATION 2

 The level settings may be changed by simply over-typing the existing
 data and then pressing ENTER.

 To set a single level enter the number. To set consecutive levels
 enter the first and last levels separated by a hyphen.

 Erased or blank fields will redisplay the current level settings as
 known to CICS. These will be the levels set when ENTER was last pressed.

 All levels may be set by entering ALL and none by entering OFF.

 To provide identical level settings for more than one component:
  - type the setting(s) for the first component, do not press ENTER.
  - type  =  in fields where you want this setting duplicated.

 NOTE: Although levels up to 32 may be set for most components, only
       levels 1,2 and 3 are currently used.


PF 3=End  7=Back  8=Next  ENTER=End
```

Figure 192. CETR transaction: Component trace HELP panel (3)

```
CETR            Help: CICS Component Trace

 (3) MEANINGS OF COMPONENT ABBREVIATIONS.

 AP . . . Application domain            IE . . . ECI over TCP/IP domain
 BA . . . Business Application Manager  II . . . IIOP domain
 BM . . . Basic Mapping Support         IS . . . ISC
 BR . . . Bridge                        KC . . . Task Control
 CP . . . CPI-C interface               KE . . . Kernel
 DC . . . Dump compatibility layer      LC . . . Local Catalog domain
 DD . . . Directory manager             LD . . . Loader domain
 DH . . . Document Handler domain       LG . . . Log Manager domain
 DM . . . Domain Manager domain         LM . . . Lock Manager domain
 DP . . . Debugging Profiles domain     ME . . . Message domain
 DS . . . Dispatcher domain             MN . . . Monitoring domain
 DU . . . Dump domain                   NQ . . . Enqueue Manager
 EI . . . Exec interface                OT . . . Object Transaction domain
 EJ . . . Enterprise Java domain        PA . . . Parameter Manager
 EM . . . Event Manager domain          PC . . . Program control
 FC . . . File control                  PG . . . Program Manager domain
 GC . . . Global Catalog domain         PI . . . Pipeline Manager domain
 IC . . . Interval control              PT . . . Partner Management

 PF 3=End  7=Back  8=Next  ENTER=End
```

Figure 193. CETR transaction: Component trace HELP panel (4)

```
CETR            Help: CICS Component Trace

 (4) MEANINGS OF COMPONENT ABBREVIATIONS.

 RI . . . RMI                          WB . . . Web Domain
 RM . . . Recovery Manager             XM . . . Transaction manager
 RX . . . RRMS domain                  XS . . . Security domain
 RZ . . . Request Streams
 SC . . . Storage control
 SH . . . Scheduler Services domain
 SJ . . . JVM domain
 SM . . . Storage manager domain
 SO . . . Sockets domain
 ST . . . Statistics domain
 SZ . . . Front End Prog Interface
 TC . . . Terminal control
 TD . . . Transient data
 TI . . . Timer domain
 TR . . . Trace domain
 TS . . . Temporary storage
 UE . . . User exit interface
 US . . . User domain

 PF 3=End  7=Back  8=Next  ENTER=End
```

Figure 194. CETR transaction: Component trace HELP panel (5)

```
CETR            Help: CICS Component Trace

 (5) FUNCTION KEY USE ON THE COMPONENT TRACE PANELS.

         Key     Name       Function

         ENTER   Change     Execute any Changes.
         PF1     Help       Display this help information.
         PF3     Quit       Return to main Trace Control panel.
         PF7     Backward   Scroll backward one page.
         PF8     Forward    Scroll forward one page.
         PF9     Messages   Display any messages created,
                            these can be warning
                            (informative) messages, or errors.









 PF 3=End  7=Back  ENTER=End
```

Figure 195. CETR transaction: Component trace HELP panel (6)

HELP panels for terminal and transaction trace

If you press the help key (PF1) while viewing the terminal and transaction trace screen, the following panels are displayed:

```
CETR                Help: CICS Terminal and Transaction Trace

 FUNCTION OF THE TERMINAL AND TRANSACTION PANEL.

  This panel has two main functions.

  (1) To allow STANDARD/SPECIAL tracing to be set for tasks to be run from
      particular terminals or with particular transaction ids.  It is also
      possible to suppress tracing for particular transaction ids.
      STANDARD tracing tasks use the STANDARD component flags.
      SPECIAL tracing tasks use the SPECIAL component flags.
      See the help information for the component panel for more detail.

  (2) To allow the setting of VTAM EXIT and ZCP tracing.

  The current settings will be displayed if transaction id, terminal id
  or netname are specified without other values.

  Further help information follows:
         1. VTAM EXIT trace.
         2. ZCP trace.
         3. Function key use on the terminal and transaction panel.

PF 3=End  8=Next  ENTER=End
```

Figure 196. CETR transaction: Terminal and transaction HELP panel (1)

```
CETR                Help: CICS Terminal and Transaction Trace

 (1) VTAM EXIT trace.

     CICS VTAM EXIT tracing consists of entries on GTF made by the CICS
     supplied VTAM exits that run asynchronously with the rest of CICS.
     The exits that are related directly to a particular terminal have their
     tracing activated by specifying the terminal id and/or netname and
     setting ON in the Terminal VTAM Exit Trace field.

     There are also non-terminal related exits for which tracing is enabled by
     specifying SYSTEM in the VTAM Exit override field.  A setting of ALL in
     the exit override field will turn on exit tracing for all non-terminal
     exits, and for all terminal exits for all terminals.

 NOTES:
  (a) Setting trace for any terminal related exit will also turn on trace of
      all non-terminal related exits.
  (b) CICS GTF tracing need not be STARTED (main panel) to use this facility.
  (c) An unknown netname is assumed to be an autoinstall terminal.
  (d) Pressing PF6 cancels all tracing from all exits.
  (e) Connection name in the termid field causes setting of all sessions.

PF 3=End  7=Back  8=Next  ENTER=End
```

Figure 197. CETR transaction: Terminal and transaction HELP panel (2)

```
CETR                Help: CICS Terminal and Transaction Trace

 (2) ZCP TRACE

     ZCP trace is the activate scan trace facility.  Turning it on
     for a particular terminal means that the activate scan routine
     of the terminal control program (DFHZCP) will start to monitor
     the activate scan queue for the Terminal Control Table Terminal
     Entry (TCTTE) for that terminal.

     When the TCTTE is found, indicating that there is activity on the
     terminal, a trace entry is made for the activate scan ZCP routine.









PF 3=End  7=Back  8=Next  ENTER=End
```

Figure 198. CETR transaction: Terminal and transaction HELP panel (3)

```
CETR                Help: CICS Terminal and Transaction Trace

 (3) FUNCTION KEY USE ON THE TERMINAL AND TRANSACTION PANEL.

         Key     Name       Function

         ENTER   Change     Execute any Changes.
         PF1     Help       Display this help information.
         PF3     Quit       Return to main Trace Control panel.
         PF6     Cancel     Cancel all exit tracing. This turns off
                            VTAM EXIT tracing for all terminals and
                            for all the non-terminal EXITs.
         PF9     Messages   Display any messages created.









PF 3=End  7=Back  ENTER=End
```

Figure 199. CETR transaction: Terminal and transaction HELP panel (4)

HELP panels for JVM trace options

If you press the help key (PF1) while viewing any of the JVM trace options screens, the following panels are displayed:

```
CETR              Help: JVM Trace Options

    FUNCTION OF THE JVM TRACE OPTIONS PANEL

     The JVM Trace Options panel is used to inquire and set the STANDARD and
     SPECIAL trace levels for tracing the JVM in the CICS system. It also
     allows setting of the JVM Trace Options used for each trace level.

     There are 4 trace levels defined, LEVEL 0, LEVEL 1, LEVEL 2 and USER.
     LEVEL 0, LEVEL 1 and LEVEL 2 are intended to correspond to trace levels
     0, 1, and 2 for JVMs, and the USER level can be used to specify deeper
     levels of tracing or complex trace options.

     The default value for these options can be overridden by SIT parameters:
     JVMLEVEL0TRACE, JVMLEVEL1TRACE, JVMLEVEL2TRACE and JVMUSERTRACE.

    Further help information follows:
          1. Data entry and alteration
          2. Function key use on the JVM Trace Options panels.

  PF 3=End  8=Next  ENTER=End
```

Figure 200. CETR transaction: JVM trace options HELP panel (1)

```
CETR            Help: JVM Trace Options

   (1) DATA ENTRY AND ALTERATION

   The fields are spread over 2 screens, Level 0 and 1 on the first and
   Level 2 and User on the second screen.  PF7 and PF8 are used to move
   between them.  The page may only be switched when the displayed page
   contains no errors.

   For each level of tracing there are 2 ON / OFF fields to set tracing
   for STANDARD and SPECIAL transactions.

   There is a 240 character "free-form" field to set the JVM Trace Options
   to use.  The meaning of these options is defined by documentation for
   the JVM.  PF6 can be used to reset all the Trace Option fields
   to the CICS defined default values.

   Changes to the ON / OFF fields are validated when the current page
   is changed, but all values are only set within CICS when ENTER
   is pressed. Until this time it is possible to quit from the program
   (PF3) without altering internal values.


  PF 3=End  7=Back  8=Next  ENTER=End
```

Figure 201. CETR transaction: JVM trace options HELP panel (2)

```
CETR            Help: JVM Trace Options

   (2) FUNCTION KEY USE ON THE JVM TRACE OPTIONS PANELS.

          Key     Name       Function

          ENTER   Change     Execute any Changes.
          PF1     Help       Display this help information.
          PF3     Quit       Return to main Trace Control panel.
          PF6     Reset      Set all Option strings back to
                             default values.
          PF7     Backward   Scroll backward one page.
          PF8     Forward    Scroll forward one page.
          PF9     Messages   Display any messages created,
                             these can be warning
                             (informative) messages, or errors.









  PF 3=End  7=Back  ENTER=End
```

Figure 202. CETR transaction: JVM trace options HELP panel (3)

Chapter 21. CIDP - inactivate debugging profiles

Transaction CIDP is a utility transaction which you can use to inactivate all debugging profiles in your system. For example, you might choose to run the transaction at the end of the normal working day to ensure that no active profiles remain in a system that is unattended.

You can inactivate all the debugging profiles in your system in the following ways:

Run transaction CIDP from a 3270 terminal
: There are no input parameters. The transaction issues a terminal message and a console message which tell you how many debugging profiles have been inactivated.

Invoke program DFHDPIN
: You can link to program DFHDPIN from another program, or include it in the second phase Program List Table (PLT). There is no input commarea. The program issues a console message which tells you how many debugging profiles have been inactivated.

For other operations on debugging profiles, use the application debugging profile manager.

© Copyright IBM Corp. 1982, 2011

Chapter 22. CIND—in-doubt testing tool

Use CIND to:

- Test the effect of in-doubt failures on application programs
- Produce shunted units of work (UOWs) to test programs that use SPI commands to inquire on, and change the characteristics of, shunted UOWs
- Unshunt UOWs that have been shunted using CIND.

Note: For a description of the concepts involved in the synchronization of distributed applications, including in-doubt periods and shunted UOWs, see the *CICS Intercommunication Guide*.

CIND can operate in a standalone system; it provides a way of changing the UOW so that it is distributed, and of causing a failure during syncpoint processing in the in-doubt window.

It can be used to cause a failure in syncpoint processing as follows:

- CIND changes the UOW to be distributed by adding itself to the UOW as the syncpoint initiator and coordinator. This has the effect of simulating a link to a remote CICS system that has assumed the syncpoint coordinator role and that has sent a syncpoint PREPARE request to CICS. This in turn forces CICS to assume the role of a syncpoint subordinate. Being a syncpoint subordinate means that during the two-phase commit syncpoint protocol, CICS itself becomes in-doubt as to the outcome of the UOW. While CICS is in-doubt, CIND simulates a link failure to the syncpoint coordinator.
- If the TRANSACTION resource definition specifies WAIT(YES), the failure in the in-doubt window causes CICS to shunt the UOW.
- If the TRANSACTION resource definition specifies WAIT(NO), a forced decision is taken to commit or back out the UOW, as defined by the TRANSACTION ACTION option. In this case, the UOW is not shunted, but is forced to back out or commit without consulting the coordinator of the distributed UOW.

You can also use CIND to shunt a UOW that is genuinely distributed across multiple CICS systems. It must be activated on the CICS system where the syncpoint is initiated, so that it can assume the role of syncpoint initiator and coordinator before any other CICS systems try to do so. In particular, CIND should not be activated for CICS mirror transactions unless the mirror can be guaranteed to be the syncpoint initiator (that is, a mirror running a server transaction that has been DPLed to specifying SYNCONRETURN).

When active, CIND forces all transactions defined to be in a new transaction class, DFHTCIND, to fail in-doubt when they reach syncpoint.

Note: You cannot use CIND on internal CICS system transactions. If you change a system transaction so that it is defined in transaction class DFHTCIND, CIND detects that it is to be used with a CICS system task, and issues message DFHIN1014 to CSMT. The system task continues to run and CIND does not cause it to fail in-doubt when it reaches syncpoint.

The required resource definitions for CIND are provided in group DFHINDT, which is a member of group list DFHLIST.

© Copyright IBM Corp. 1982, 2011

Running the CIND indoubt tool

Run the CIND indoubt tool from a terminal to test the effects of indoubt failures on application programs. You can also use this transaction to produce shunted units of work and change the characteristics of units of work.

When active, the tool forces all transactions that are defined in the transaction class, DFHTCIND, to fail in doubt when they reach a synchronization point. You cannot use CIND on internal CICS system transactions.

1. Change the transaction class of the transactions that you want to go indoubt to DFHTCIND.
2. Enter `CIND ON`. This command activates the tool and enables a task-related user exit for the task start. You get a message confirming that the tool is active.

   For every transaction that starts, the task-related user exit inquires on the transaction class. If the class is DFHTCIND, the task-related user exit issues a call to the CICS recovery manage to add a link to CIND, acting as a CICS coordinator system called DFHINDSP.
3. Run you transactions. If the transactions update recoverable resources and the TRANSACTION resource specifies the attribute WAIT(YES), they fail in doubt. The transaction fails with abend code ASP1 and the unit of work is shunted. You can inquire on the shunted units of work.

   A transaction that updates nonrecoverable resources, or that issues only READ requests, fails with abend code ASPR. The unit of work is not shunted because it is read-only.
4. Enter `CIND OFF`. This command disables the task-related user exit and stops any transactions going indoubt. You get a message confirming that the tool is no longer active.
5. Enter `CIND RESYNC COMMIT` or `CIND RESYNC BACKOUT`. These commands simulate a coordinator CICS system reconnecting to your CICS region. The coordinator is DFHINDSP.

The CICS recovery manager unshunts all units of work that were previously shunted because of the tool. The commit or backout specifies what action the recovery manager does when the unshunt takes place. For each unit of work that is unshunted, message DFHIN1012 is sent to CSMT. This message details the unit of work status as defined by DFHINDSP and the recovery manager.

You can also use these commands to simulate what happens units of work are not synchronized. For example, if a TRANSACTION resource specifies the attributes WAIT(NO) and WAITACTION(COMMIT), the unit of work is not shunted but forced to commit. If you enter a `CIND RESYNC BACKOUT` at this point, the DFHIN1012 message output shows the unit of work to be out of sync, as does message DFHRM0111.

Inquiring on the status of CIND

Type `CIND INQUIRE`. This returns the current status of CIND with one of the following messages:

```
DFHIN1003 date time applid.  The in-doubt tool
is active for DFHTCIND tranclass transactions.

DFHIN1006 date time applid.  The in-doubt tool is
not active.
```

Running the in-doubt tool using an EXEC CICS LINK command

Instead of running the in-doubt tool using a terminal and the CIND transaction, you can use the EXEC CICS LINK command to link to the in-doubt tool from an application passing a COMMAREA. For example:

```
EXEC CICS LINK PROGRAM('DFHINDT') COMMAREA(COMM)
```

where COMM is a data-area that contains the commands to pass to DFHINDT. COMM should contain one of:

```
' ON '
' OFF '
' RESYNC COMMIT '
' RESYNC BACKOUT '
```

Setting a single transaction in-doubt

If you want a single transaction to go in-doubt regardless of its transaction class, an application can link to program DFHINDAP. (No commarea is required in this case.) This causes that single transaction to go in-doubt without the need to type CIND ON. This single transaction can be resynchronized by typing CIND RESYNC COMMIT, and so on, as described in “Running the CIND indoubt tool” on page 500.

DFHINDAP works by enabling the in-doubt task-related user exit if necessary and then issuing an API call to the in-doubt task-related user exit to add the dummy coordinator. This has the advantage of allowing a test application to decide it wants to force in-doubt dynamically, after it has started. Hence a UOW, other than the first one in the transaction can be forced in-doubt.

CIND Messages

The in-doubt tool outputs audit trail DFHINxxxx messages to CSMT at the following times:

- When CIND is activated using CIND ON
- When CIND is deactivated using CIND OFF
- When a resync is initiated using CIND RESYNC COMMIT or CIND RESYNC BACKOUT
- For each UOW acted upon, when:
  - CIND adds itself as coordinator to the UOW
  - CIND simulates losing the link to the coordinator system and the UOW is shunted
  - CIND simulates losing the link to the coordinator, but the UOW is not shunted
  - CIND causes resynchronization of a UOW.

Each message shows the transaction ID, the task number, and the UOW identifier.

Chapter 23. CLDM - CICS load module map

Transaction CLDM is used to generate a CICS load module map. The load module map can be used by the z/OS hardware instrumentation service or during problem determination to help identify a program that has been loaded by the CICS loader domain.

The CICS load module map can be generated in different formats using the options shown in following syntax diagram.

pathname
Is the name of an existing USS file system path that CLDM uses to store a z/OS hardware instrumentation services CICS load module map. The path name must be an absolute pathname, that is, it must start with a forward slash (/).

Because the USS file system path name is usually lowercase when CLDM is invoked on a terminal or console that is defined with MSGCASE=UPPER, CLDM converts the path value to lowercase. If the path name is not converted to lowercase, it must be enclosed in single quotes. If the path name is mixed case, ensure that the terminal definition is appropriate and enclose the path name in single quotes.

The file name written to the specified USS file system path is generated by CLDM and has the format `SYSHISTyyyymmdd.hhmmss.asidx.jobname.CICSMAP`, where:

- `yyyymmdd` is the year, month and day that CLDM ran.
- `hhmmss` is the hour, minute and second that CLDM ran.
- `asidx` is the 4 byte hexadecimal address space ID of the CICS region.
- `jobname` is the CICS job name.

The format of records written to the USS file system are described in the z/OS System Commands manual, table 'Interpreting the information in a .MAP file from hardware data collection'. However the following conditions apply:

- CLDM generates data records only with a record type of M (Module) and C (CSECT).
- For module (Record type M) records, if the module is in an LPA region (that is MLPA, PLPA or FLPA), the memory area is always set to M.
- The long name (VOLSER and DSN) field is always blank for module records.

Note: It is the CICS region user ID and not the user ID associated with the CLDM transaction that is checked for authority to create the file in the USS file system.

sysoutclass
Is the sysout class that CLDM must use to produce a loader program map report. CICS must have the CICS spool interface active (SPOOL= YES in the SIT) for the SYSOUT operand to be effective. No other spool related attributes must be specified. You can invoke CLDM in the following ways:

© Copyright IBM Corp. 1982, 2011

- Run the transaction from a 3270 terminal. Input is received from the terminal and messages are sent to the terminal.
- CLDM can be started from a CICS API START command. Input is retrieved from the START command and messages are sent to the console.
- DFHLDMAP can be linked to. The input is obtained from the current channel in a container DFHLDM.INPUT and messages are written to the container DFHLDM.OUTPUT.

Note: This support is intended to map executable code. Literal pools may not be included in the CSECT mappings for this release.

Chapter 24. CLER - Language Environment runtime options

The CLER transaction is provided by Language Environment and is not shipped as part of the CICS Transaction Server product.

Use the CLER transaction to display the Language Environment runtime options. You can also modify a subset of the options. If you want to print the runtime options, you can write them to the CESE queue.

CLER is a conversational transactional that presents the display panel shown in Figure 203.

```
 CLER                                                        CICS IYCLZCFE

              LANGUAGE ENVIRONMENT REGION LEVEL RUNTIME OPTIONS


TYPE IN YOUR CHOICES.

RUNTIME OPTION     CHOICE      POSSIBLE CHOICES


ALL31          ==>  ON          ON, OFF
CBLPSHPOP      ==>  ON          ON, OFF
CHECK          ==>  ON          ON, OFF
INFOMSGFILTER ==>  OFF         ON, OFF
RPTOPTS        ==>  OFF         ON, OFF
RPTSTG         ==>  OFF         ON, OFF
TERMTHDACT     ==>  DUMP        QUIET,MSG,TRACE,DUMP,UAONLY,UADUMP,UAIMM,UATRACE
TRAP           ==>  OFF         ON, OFF


WHEN FINISHED, PRESS ENTER.
```

Figure 203. CLER run-time options transaction

Note:

1. The CLER transaction uses basic mapping support (BMS) and therefore this function is available only through those terminal devices that are supported by BMS.
2. The transaction, program, and mapset resource definitions that you require to run CLER are in the CEE group. They are added to the CSD automatically during CICS installation, as part of the DFHCOMDS job. They are also supplied in the Language Environment SCEESAMP library.

For information about how to use the CLER transaction, see the *z/OS Language Environment Debugging Guide and Run-Time Messages*.

© Copyright IBM Corp. 1982, 2011

Chapter 25. CMAC - messages and codes display

Use CMAC to display information about CICS messages and codes.

To use the CMAC transaction:

- Add the DFHCMAC group on the CICS system definition (CSD) file to your initialization list (DFHLIST)
- Add the data set name of the CICS messages file to the definition of the DFHCMACD file within the DFHCMAC group.

For information about how to carry out these actions, see the *CICS Resource Definition Guide*.

Note:

1. This transaction is not valid for the MVS console.
2. The information is available only in English.

You start the transaction by typing one of the following:

1. **CMAC**
2. **CMAC abcd** where 'abcd' is the abend code.
3. **CMAC xxxx** where 'xxxx' is the message number only.
4. **CMAC yyxxxx** where 'yy' is the component id and 'xxxx' is the message number.
5. **CMAC DFHxxxx** where 'xxxx' is the message number.
6. **CMAC DFHyyxxxx** where 'yy' is the component id and 'xxxx' is the message number.

If you use method 1, you get the display shown in Figure 204.

```
DFHCMC01              Display On-line Messages and Codes


Type the required message identifier, then press Enter.



     Component ID ===>  (for example, TC for Terminal Control
                                      FC for File Control, etc.)
                        This field is required for Messages in the
                        form DFHxxyyyy, Where xx is the Component ID.

     Message Number =>  (for example, 1060, 5718, or Abend Code
                        such as ASRA, etc.)









F3=Exit to CICS
```

Figure 204. CMAC transaction: initial screen

© Copyright IBM Corp. 1982, 2011

You tell CICS which message you want by typing the component ID and the message number. If the message does not have a component ID, leave that field blank.

The display then gives information under the headings:

Explanation
An explanation of the cause of the message.

System action
The actions the system is taking.

User response
What actions you can or cannot take.

Destination
Where to see the message. This can be the console, the end-user terminal, or a transient data queue.

Module
The name of the module reporting the error.

For example, to display information about message DFH5130, which does not have a component ID, type:

```
CMAC 5130
```

You get the following display:

```
DFH5130   UNABLE TO LOCATE MODULE DFHCICS.  PRIMARY
          CSD NOT INITIALIZED.
EXPLANATION:  The DFHCICS module is missing from the library.
SYSTEM ACTION:  Processing of the INITIALIZE command is
terminated.
USER RESPONSE:  Ensure that the DFHCICS module is present in
the library.
DESTINATION:  SYSPRINT
MODULE(S):  DFHCSDUP
F3=CANCEL
```

Figure 205. CMAC transaction: 1st example screen

To display the message CE3528, type a component ID of CE and a message number of 3528. You get the following display:

```
DFHCE3528 Signon failed during SECLABEL checking.


EXPLANATION:   The signon request has failed because the
external security manager (ESM) detected a critical error.

SYSTEM ACTION:  The signon transaction terminates.

USER RESPONSE:  Refer to message DFHSN0108 on the CSCS log
for the information and actions necessary to resolve this
problem.

DESTINATION:  Terminal End User

MODULE(S):  DFHSNP




F3=CANCEL
```

Figure 206. CMAC transaction: 2nd example screen

If you use methods 2 through 6 (listed above), the initial screen is bypassed and you move directly to the message information screen.

Chapter 26. CMSG - message switching

Use CMSG to send messages from your terminal to one or more destinations.

The destination may be:

- Another terminal
- Another terminal, but only when a specified operator is signed on at that terminal
- Another terminal, but only when the operator signed on at that terminal is in a specified class
- Another operator (CICS locates the operator, if currently signed on, and sends the message to that operator's terminal)
- All terminals
- All terminals with operators signed on in specified classes
- Any combination of these destinations.

You can specify that the message is to be sent at some future time. If the message cannot be sent, your terminal is notified.

In addition, standard routing lists can be provided using the terminal list table (TLT), which can include terminal and operator identifiers. Multiple routing lists can be specified at the time the message is entered, and individual destinations can be deleted from, or added to, the lists at this time.

CICS message switching is provided by a service program invoked by a user-defined transaction identifier that must be defined to CICS. This transaction identifier can be CMSG or any user-defined 4-character transaction identifier. CMSG is the transaction identifier assumed throughout this book.

You can extend message switching to include predefined messages and even complete message-switching transaction input formats.

The terminals from which message switching is initiated, and all terminals of message destinations, must support BMS.

Routed messages can be directed to a terminal, including the transaction terminal, that supports partitions. However, such messages reset the terminal to "base" state.

Sending a message

To send a message to another terminal or terminals, enter the transaction identifier, CMSG, followed by the applicable options, as shown in the syntax diagram "CMSG command" on page 515.

Full details of the options are given under "CMSG command options" on page 516.

After entering CMSG, you must include at least the message text, a destination (ROUTE, or OPCLASS, or both), and the option SEND (which you can abbreviate to S). For example:

```
CMSG 'GOOD MORNING',R=ALL,S
```

sends the message 'GOOD MORNING' to all terminals.

© Copyright IBM Corp. 1982, 2011

If all is well, your message is sent, and the response

```
MESSAGE HAS BEEN ROUTED
```

is returned to your terminal.

This response is received for the normal completion of a message-switching transaction. It means that the message has been successfully routed to its destination but does not reflect any delivery status.

You can use backspace characters in your input.

New-line characters (or carriage returns on a hard-copy terminal) that are not part of the message are changed to commas, unless they appear at the beginning or end of the input, in which case they are deleted. In this way, you can use NL characters as option separators, instead of commas. For example, on a display device, you could enter:

```
CMSG R=(T001,T002)#
T=1500#
'THE TIME IS 1500 HOURS'#
S@
```

where the ENTER key.

You can enter the options, except SEND and CANCEL (one of which must be entered last), in any order after CMSG and one or more spaces. You can abbreviate every option, except CANCEL, to a single character, and you can include one or more options in a single input. If you specify two or more options, you must follow each by a comma, except the last, which must be followed by a space or by an end-of-data (EOD) condition indicating the end of your input. Any data following a space in the same input is ignored.

To mark the end of a single input:

- Press the ENTER key on display devices such as the IBM 3270
- Press carriage return for hardcopy terminals such as the IBM 2741.

To maintain the conversation with a terminal, the status of all inputs, except the final input which includes the SEND option, is saved in a single temporary-storage record.

A message-switching transaction remains conversational until ended. This means that all input (even other transaction identifiers) is passed to the message-switching program. You must end a message-switching transaction (possibly involving multiple inputs) before you can start a different transaction.

Canceling a message

You can cancel your current message-switching transaction by entering:

- CANCEL as the last 6 characters of your input. You get the response:

  ```
  TERMINATED BY OPERATOR.
  ```

- CMSG as your first 4 characters, followed by end-of-data (EOD). You get the response:

  ```
  NO INPUT - REENTER.
  ```

- CMSGx as your first 5 characters, where x is any character other than a space or a C. You get the response:

```
SPACE MUST FOLLOW TRANSID.
```

- CMSG as your first 4 characters followed by a space and EOD. This starts a new message-switching transaction. You get the response:

  ```
  CONTINUE INPUT.
  ```

The first three examples in this list terminate the conversation and allow you to enter other transactions.

Message-switching continuation

The message-switching transaction appears to be conversational. Internally, a new task is created to process each message switching input.

If a task requires more information to complete the transaction, the current status of the transaction is saved in temporary storage, a response is sent to your terminal, and the task terminates with a requirement that the next task started be a message-switching task.

The saving of status replaces any previously saved status for any message-switching transaction from that terminal.

If a message-switching transaction completes successfully, or you have canceled it, an appropriate response is sent to your terminal, the status of the transaction is not saved, and the task terminates with no transaction restart specified. The saved status is always used by a message-switching task to continue a conversational input from your terminal. You can recall the saved status by entering the transaction identifier to which you have added C (for example CMSGC). You can include additional input, but it must follow CMSGC with no intervening spaces.

This method is useful because you can enter CANCEL to bypass errors in the current input and yet not allow the conversation to continue. You can then enter, for example CMSGC and then enter EOD, which puts the terminal in the conversation that prevailed at the time of the most recent input that has not been canceled or has not resulted in a message being routed. At this point you could also add an option if desired, for example CMSGCT=1030. (Note that there are no spaces between the transaction identifier, the C, or the following data.)

For example, enter:

```
CMSG R=(T40,T41)
```

and you receive:

```
R OK CONTINUE INPUT,
T=1500
RT OK CONTINUE INPUT
'THERE WILL BE ..... AT THE
RT OK CONTINUE MSG
MAIN NIO;FOMH PM VS;OGPTMOS SBR, CANCEL (see Note)
TERMINATED BY OPERATOR
CMSGCMAIN BUILDING ON CALIFORNIA AVENUE
RT OK CONTINUE MSG
IN ROOM 407',SEND
MRTS OK MESSAGE HAS BEEN ROUTED
```

Note: You can cancel and reenter instead of trying to correct the input. This terminates the entire transaction. To save the previously entered input, enter CMSGC to restart at the previously saved status point and continue the message.

Your terminal is free between a message-switching response and the next input, and if it is in TRANSCEIVE status, an automatically initiated task can be started on your terminal before you can continue the message-switching transaction.

Upon completion of this new task, even if it required a response from you, the message-switching transaction continues as if nothing had intervened. However, if you receive a response indicating that the intervening task has issued a RETURN, you must enter the transaction identifier followed by C to force continuation of the previous message-switching transaction.

Message delivery

You can deliver a message to a destination if the following conditions are met:

- The specified delivery time has been reached
- The terminal is INSERVICE, and either:
  - The opid of the operator signed-on matches the requested opid if specified in the ROUTE option

    or
  - The class of the operator signed-on matches any class specified by the OPCLASS option.

The processing and paging status of a terminal, which also affects delivery of eligible messages, is initially specified on the CEDA DEFINE TYPETERM command. You can change the status using the CEMT, CEST, and CEOT transactions.

When the processing status is RECEIVE or TRANSCEIVE, a message is sent automatically to the terminal when no transaction is attached to it.

When the processing status is TRANSACTION, you must request message delivery by entering a paging command, see Chapter 30, “CSPG - page retrieval,” on page 553.

A message is formed into one or more “pages”, the size of which is predefined for a destination by the system programmer. The terminal processing status determines the delivery of the first or only page of a message. Succeeding pages, if any, are sent according to the paging status of the terminal as defined by the system programmer.

When the terminal is in PAGE status, you must request succeeding pages through a paging command.

When the terminal is in AUTOPAGE status, the complete message is printed.

Subsystems and terminal systems

With the exception of any specific details relating to the ROUTE option (see CMSG command options, ROUTE), all other information on the use of the following subsystems in message switching is given in the appropriate IBM subsystem guides:

- IBM 3600 Finance Communication System (VTAM), SC33-0233
- IBM 3650 Retail Store System, SC33-0234
- IBM 3770 Data Communication System, SC33-0235
- IBM 3790 Communication System, SC33-0236.

IBM 3270 display devices

The responses to 3270 family display devices are designed to fit on a single 40-character line of a Model 1 display. To provide more space for the response, the indicators that precede the response do not have the spaces between them that appear on non-3270 terminals.

Responses to the 3270 family display devices do not erase what is displayed on the screen and are displayed in the last 40 character positions of the line.

After a message-switching transaction has been started, you can press the CLEAR key if you want to continue from an empty screen. A CLEAR/CANCEL OR CONT xxxxx response is displayed, where xxxxx is either INPUT, ROUTE, or MSG. You can continue the transaction or, if you press the CLEAR key a second consecutive time, you can cancel the transaction. A better method of erasing the screen is to press the ERASE INPUT key, because this does not cause an interaction with the processor.

Because positions on the screen that are not keyed are not received, you can position and display data on the screen in various ways. For example, you can start each option on a new line of the display screen. You must include a comma between consecutive options. Also, if you want to start a new line within the message text you should enter a field mark character to indicate the end of the current line. All field mark characters within a message (including the first character of the message) are interpreted as NL characters (there is no corresponding key on 3270 family display devices) and cause correct positioning of the message on all other terminals.

IBM 3767 Communication Terminal

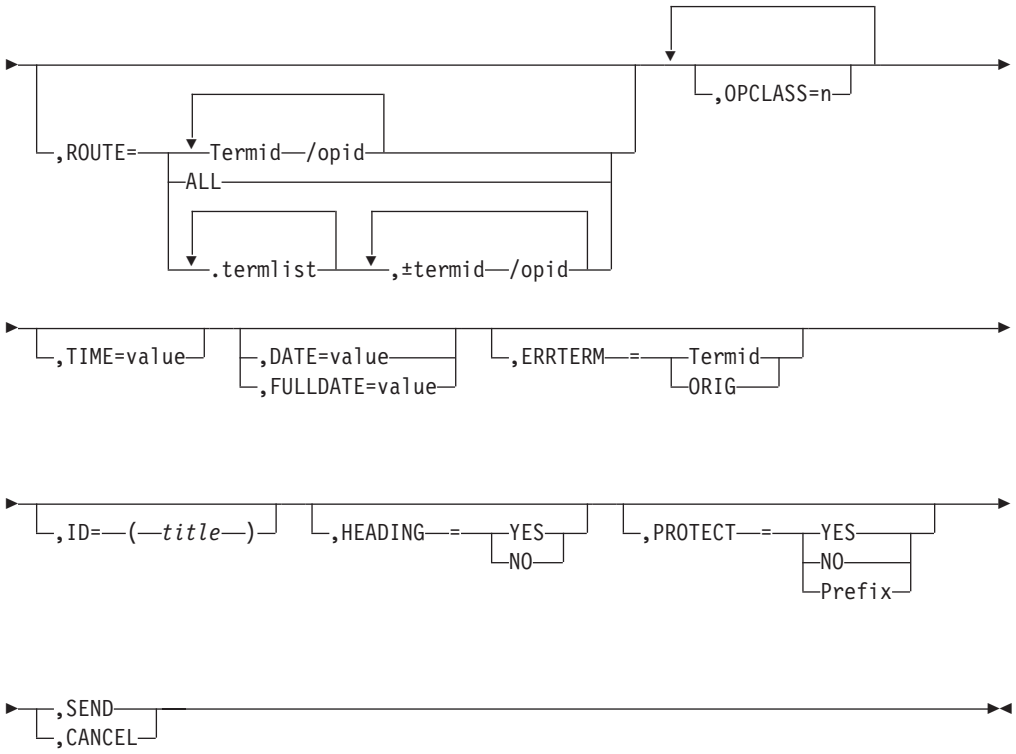
The keyboard/printer unit of the 3767 is eligible for entering and receiving messages.

Sequential terminal

On a sequential (BSAM) terminal, the CMSG message-switching facility remains conversational until a message-switching transaction is completed or canceled. Variable conditions requiring a response, such as operators not being signed on, can be handled easily from an interactive terminal. These conditions, however, cannot always be anticipated, and the message-switching task continues reading input and issuing error messages until a valid response is received. It is possible that an entire sequential input stream could be consumed in such a process. You are advised to ensure that a CMSG CANCEL record appears in the sequential input following each message-switching transaction that could possibly require an unanticipated reply. This is strongly recommended following the final message-switching transaction.

CMSG command

CMSG



CMSG command options

The message-switching options are listed below. Except for CANCEL, you can specify the first letter of each option instead of the entire option.

CANCEL
specifies that the current input is to be ignored and institutes a **non**-conversational status between the terminal and the message-switching transaction. CANCEL must be the last 6 characters of the input. CANCEL is also effective within a message.

DATE=value
The date on which you want your message to be delivered. It can be specified in any of the following forms:

yy.ddd
year (00–99) and day (001–366).

mm/dd/yy
month (01–12), day (01–31), and year (00–99).

mm/dd
month (01–12) and day (01–31).

+d number of days (0–4).

The first three of these forms provide ways of specifying absolute dates, with the year (where used) in a 2-digit format. For example, if the current system date is in the year 1997, January 31 1997 could be specified as 97.031, 01/31/97, or 01/31. In this last case, the year of the current system date is assumed to be the year for delivery of the message.

Note: References to "system date", "system time", "system year" and so on mean the date time or year as would be returned by EXEC CICS ASKTIME.

If DATFORM=DDMMYY was specified in the CICS system initialization parameters, enter the second and third of these as dd/mm/yy or dd/mm.

- If you want to specify an absolute date with the year in a 4-digit format, then use the FULLDATE parameter.

The fourth form allows you to specify a number of days from today. For example, a value of DATE=+3 (or D=+3) means that the message is to be transmitted 3 days from today. The number must be in the range 0–4. DATE=+d entries are not accepted when the system time is between 2330 and 0030, (to avoid confusion at or near midnight). If you use this form of the command within 30 minutes of midnight, the following error message is issued:

```
+DATE INVLD FROM 2330 to 0030
```

You can also specify a time for message delivery using the TIME= option, which is described CMSG command options, SEND. The effects of TIME= and DATE= together are as follows:

- If you specify neither a time nor a date, the message is transmitted as soon as the receiving terminal is free.
- If you specify a time but no date, the message is transmitted at the specified time today. For example, if the time now is 09.00 and you specify TIME=0930, or TIME=+30, the message is transmitted at 09.30 today.
- If you specify a date but no time, the message is transmitted at the current system time on the specified date. For example, if the time now is 10.30 and you specify DATE=+2, the message is transmitted at 10.30 in two days' time.
- If you specify both a date and a time, the message is transmitted at the specified time on the specified date. For example, if you specify DATE=07/29/98 and TIME=1130, the message is transmitted at 11.30 on 07/29/98.

Note:

1. In all cases, the delivery time that you request must be less than 100 hours from the beginning of the current day. This means that the delivery time can never be later than 03.59 on the fourth day from the current day.
2. When processing date options entered in the form yy.ddd, mm/dd/yy and dd/mm/yy, CMSG operates a 'sliding 50 year window' to establish whether the year is in this century, the previous century or the next century. The two digit year is initially assumed to be in the same century as the current date. If this assumed year is more than 50 years in the past or more than 50 years ahead, it is adjusted accordingly. For example, if today's date is the 31st December 1997, the following DATE options are handled as follows:
   - DATE=99.001 is initially assumed to be the year 1999. Since it is within 50 years of the system year, the year 1999 is determined to be the delivery date for the message.
   - DATE=00.001 is initially assumed to be the year 1900. Since this year is more than 50 years ago, the delivery date is established as the year 2000.

   In both of these examples above, the delivery date is not accepted and the message 'DATE TOO FAR IN FUTURE' is displayed.

Note that the FULLDATE operand allows a four digit year to be specified, and removes any possible ambiguity when using the DATE operand.

ERRTERM

"termid" is the identifier of the terminal to which notification is to be sent if the message is purged because it is undeliverable.

ORIG is a way of specifying the identifier of the originating terminal.

Note: A message is considered undeliverable to a destination if it cannot be delivered within a specified interval after the requested delivery time. This interval is specified by the system programmer. If no interval is specified, no action is taken for undelivered messages, and the ERRTERM option has no effect.

If PRGDLAY is specified in the system initialization table (DFHSIT), the transient data destination CSMT is notified of the number of undeliverable messages purged for a terminal. In addition, if ERRTERM is entered, the specified terminal is notified of the message number, title identifier, and destination of the message.

FULLDATE=value

The FULLDATE option is similar to the DATE option, but it requires a four-digit year to be entered. It specifies the date on which you want your message to be delivered. It can be specified in any of the following forms:

yyyy.ddd
year (0000–9999) and day (001–366).

mm/dd/yyyy
month (01–12), day (01–31), and year (0000–9999).

mm/dd
month (01–12) and day (01–31).

+d number of days (0–4).

The first three of these forms provide ways of specifying absolute dates, with the year (where used) in a 4-digit format. For example, if the current system date is in the year 1997, December 31 1997 could be specified as 1997.365, 12/31/1997 or 12/31. In this last case, the year of the current system date is assumed to be the year for delivery of the message.

(If DATFORM=DDMMYY was specified in the CICS system initialization parameters, enter the second and third of these as dd/mm/yyyy or dd/mm).

The fourth form allows you to specify a number of days from today. For example, a value of FULLDATE=+3 (or F=+3) means that the message is to be transmitted 3 days from today. The number must be in the range 0–4. FULLDATE=+d entries are not accepted when the system time is between 2330 and 0030, (to avoid confusion at or near midnight). If you use this form of the command within 30 minutes of midnight, the following error message is issued:

```
+DATE INVLD FROM 2330 to 0030
```

You can also specify a time for message delivery using the TIME= option, which is described in CMSG command options, SEND. The effects of TIME= and FULLDATE= together are as follows:

- If you specify neither a time nor a date, the message is transmitted as soon as the receiving terminal is free.

- If you specify a time but no date, the message is transmitted at the specified time today. For example, if the time now is 09.00 and you specify TIME=0930, or TIME=+30, the message is transmitted at 09.30 today.
- If you specify a date but no time, the message is transmitted at the current system time on the specified date. For example, if the time now is 10.30 and you specify FULLDATE=+2, the message is transmitted at 10.30 in two days' time.
- If you specify both a date and a time, the message is transmitted at the specified time on the specified date. For example, if you specify FULLDATE=07/29/1998 and TIME=1130, the message is transmitted at 11.30 on 07/29/1998.

Note: In all cases, the delivery time that you request must be less than 100 hours from the beginning of the current day. This means that the delivery time can never be later than 03.59 on the fourth day from the current system date.

HEADING
specifies heading information. You can use H or HEADING in place of HEADING=YES.

YES Specifies that the current time, date, and identifier of the originating terminal is to precede the message text.

NO causes a previous heading request to be ignored.

ID=(*title*)
title specifies the title (maximum length 62 characters) to be associated with the message.

See Chapter 30, "CSPG - page retrieval," on page 553 for commands to request a display of the titles of all messages queued for immediate delivery to that terminal.

MSG=message
"message" is the text of the message to be sent. The keyword MSG and the equal sign are optional. You must enclose the text within single quotation marks. A single quotation mark to be included as part of the message must be represented by a pair of single quotation marks. The message may be continued across multiple consecutive inputs.

If the ending single quotation mark is omitted, the entire input is treated as part of the message and a request to continue the message is sent to the terminal. The entire transaction may be canceled, or alternatively, options previously entered for this transaction may be saved by entering a single quotation mark followed by a comma to terminate the MSG option. The correct message can then be reentered; the previous incorrect message being ignored.

A single quotation mark at the end of data in a MSG option means either the end of the MSG option, or the first of a pair of single quotation marks indicating that a single quotation mark is to be included as part of the message.

In this situation, the response to the terminal is:

```
CONTINUE INPUT OR MSG
```

If the first character of the next input is a single quotation mark, it is treated as the second of a pair of single quotation marks and the message is continued. Any character other than a single quotation mark causes the message to be complete, and that character is treated as the first character of a new option.

New-line (NL) characters within the message are kept. (If the first character is a new-line character, it is deleted.) This allows the operator to enter `M='` and then carriage return (CR) or the equivalent of CR, to begin entering the message text at the left margin. The first CR is deleted. Additional CRs may be entered if blank lines are desired at the top of the transmitted message.

Note: If the HEADING option is specified, these blank lines appear between the heading (time, date, and originator's terminal identifier) and the message.

With NL processing, the delivered message is positioned at the left margin. If an unformatted message, or a line within a formatted message, exceeds the line width defined for the receiving terminal, sentences are split between words for any line exceeded.

OPCLASS

One or more numbers, each of which can be in the range 1–24, that define the operator classes that must be signed on before a message can be delivered. If more than one number is specified, the list must be enclosed within parentheses. For example, OPCLASS=(8,2) causes the message to be sent to all terminals that currently have an operator of class 8 or 2 signed on, and to all terminals that have that operator security value specified in their installed definitions. If OPCLASS=1 is specified, the message is routed to all terminals that are in service, regardless of whether an operator is signed on or not.

If ROUTE is specified as well, the message is routed to all requested destinations, but is not eligible for delivery to a terminal unless the class of the operator signed on matches one of the numbers specified by OPCLASS. However, if a ROUTE destination is qualified by an operator identifier, OPCLASS is ignored for that destination. For more information about how ROUTE= and OPCLASS= are used together, see the description of the ROUTE option.

PROTECT

specifies message recovery for a CICS emergency restart. You can use P or PROTECT in place of PROTECT=YES.

YES Specifies that $$ is to be prefixed to the temporary storage data identifier of the stored message.

NO Specifies that a previous protect request is to be ignored. This is done by using the default prefix of **. The same method is used to omit the option altogether.

prefix

Specifies a 1-or 2-character prefix to be used for the temporary-storage data identifier of the stored message. If a single character only is specified, a $ is provided as the second character. (For example, PROTECT=T causes a prefix of T$.)

If this option is omitted, a default prefix of ** is used. ** is also the default for user application programs issuing BMS message requests where no protection is specified (REQID option omitted).

A temporary-storage table (TST) entry is needed for each prefix specified in the PROTECT option so that message recovery is effective for that prefix.

ROUTE

specifies the destinations to receive the message. For routing messages to 3600, 3770 (batch), or 3790 (batch) terminals, see "Examples of 3600 and 3770 batch destinations" on page 526.

Termid
is the identifier or identifiers of the terminals to which the message is to be routed. For example, ROUTE=(LA04,OL,SF2) routes the message to the three terminals with the identifiers LA04, OL, and SF2. If routing is performed to several terminals of the same device and map suffixes, CICS processes the message identically for all of them and the most restricting page size prevailing is used.

The length of the terminal identifier specified in a message-switching transaction must be in the range 1–4 characters, and must not contain any of the following characters:

/ slash
, comma
) right parenthesis
(left parenthesis
+ plus sign
- minus sign
***** asterisk
blank.

Note: A single message can be delivered more than once to the same terminal. For example, the instruction ROUTE=(T001,T001) causes two transmissions of a single message to terminal T001. If the destination terminal is in TRANSCEIVE status, the message appears consecutively at the terminal. If the terminal is in TRANSACTION status, the operator must request delivery of the message.

/opid
is a 1-to 3-character operator identifier preceded by a slash. The message is routed to the first terminal at which an operator with that identifier is currently signed on. For example, ROUTE=/PJ routes the message to the first terminal found (and only the first) with the operator identifier PJ currently signed on. If no such terminal is found, the sending operator is notified. The operator identifier that you specify must not contain any of the following characters:

, comma

) right parenthesis

space.

Termid/opid
is a terminal identifier qualified by an operator identifier to restrict the message delivery to the specified operator at the terminal location. For example, ROUTE=(LA04,OL/LBS,SF2) routes the message to terminals LA04 and SF2. The message is routed to terminal OL only if the operator whose identifier is LBS is signed on at that terminal.

ROUTE=(T001,T001/OP1,/OP1) causes the same message to be delivered three times to the same destination if the operator OP1 is signed on at T001.

ALL
causes the message to be broadcast to all terminals.

There is a variable limit on the number of terminals to which a message can be sent. This limit depends on a combination of factors. Significant factors are the types of terminal in use, the number of each type, and the

length of message sent. The CMSG transaction is abended with an abend code of ABMC if the limit is exceeded.

Note: If a CMSG ROUTE=ALL is issued to a large number of terminals, a task for each terminal is initiated up to the MAXTASK value. Because the tasks are single threaded, they are suspended and can give rise to an SOS condition. For guidance about avoiding this, see the *CICS Performance Guide*.

.termlist
is a 1-or 2-character terminal list table (TLT) suffix preceded by a period. For example, .H3 identifies the terminal list table DFHTLTH3. A maximum of 10 terminal lists can be specified, and the terminal lists that you specify are merged together. The entries in the terminal lists contain terminal identifiers, or operator identifiers, or both. Duplicate entries within a single TLT are kept, though entries that are duplicated among the lists are deleted. (Entries are considered duplicate if each has the same terminal identifier and operator identifier.)

Here are two examples that show the effects of merging TLTs that contain duplicate entries. For these examples, assume that terminal list table DFHTLTL1 contains T001 twice, and that DFHTLTL2 contains T001 and T001/OP1.

- If you specify ROUTE=(.L1,.L2), all entries from DFHTLTL1 are included as destinations. Duplicate entries within DFHTLTL1 are kept. All entries from DFHTLTL2 are checked for duplicates against the entries in the previously specified DFHTLTL1 and, if a duplicate is found, it is not repeated.

  The resulting destination list is T001, T001, T001/OP1.
- The order in which you specify the TLTs is significant. If you specify R=(.L2,.L1), the DFHTLTL2 entries T001 and T001/OP1 are included in the destination list. However, the two entries for T001 in DFHTLTL1 are not included because T001 is already in DFHTLT2. In this case, the resulting destination list is T001, T001/OP1.

(±*termid/opid*,...**)**
A +termid/opid adds the specified destination (if not a duplicate) to the destinations contained in the requested TLT. A -termid/opid deletes the specified destination from the requested TLT. A -termid, without an opid, deletes all destinations of that terminal (with or without operator identifier) resulting from the requested TLT. + or -termid/opid parameters affect only those entries that result from requested TLTs, and have no effect on other + or - termid/opid parameters in the same request. All TLT suffixes must be entered before any + or - parameters.

Here are some examples that show the effects of specifying both TLTs that contain duplicate entries and ± entries. For these examples, assume that terminal list table DFHTLTL1 contains T001 twice, and that DFHTLTL2 contains T001 and T001/OP1.

- ROUTE=(.L1,.L2,+T001) has the same effect as R=(.L1,.L2). The entry +T001 is not added, because it is a duplicate of an entry from DFHTLTL1. The resulting destination is T001, T001, T001/OP1.
- ROUTE=(.L1,.L2,+T001/OP1,-T001) does not add +T001/OP1 because it is a duplicate of an entry in DFHTLTL2. The -T001 causes all entries from TLTs that refer to T001 (regardless of whether they are qualified by an operator identifier) to be deleted. The message ALL ROUTE ENTRIES DELETED is issued.

If DFHTLTL2 did not contain the entry T001/OP1, the +T001/OP1 instruction would cause that entry to be added to the destination list. The -T001 instruction would not then delete the T001/OP1 entry from the list, because the effects of the + and - instructions are not cumulative: they act in isolation on the original concatenated TLTs.

- ROUTE=(.L1,.L2,-T001,+T001/OP1); the -T001 causes all entries from the TLTs that refer to T001 (including the T001/OP1 entry in DFHTLTL2) to be deleted. The +T001/OP1 entry is then added and becomes the only resulting destination. There is no duplicate because it has just been deleted.

A ROUTE option may be divided across multiple consecutive inputs. However, if it refers to a TLT, it must be completed in the same input in which it was started. An individual ROUTE parameter (termid/opid) may not be split across two inputs.

When both ROUTE and OPCLASS are specified together, OPCLASS further restricts the message transmission. For example, ROUTE=(LA04/PJL,/MGK,OL), OPCLASS=4 routes the message to terminal LA04 if the operator whose identifier is PJL is signed on. The message is also sent to the first terminal with the operator whose identifier is MGK signed on. An operator whose class is 4 must be signed on to OL before the message can be routed there. Note that the OPCLASS value is acted on only when no operator identifier is specified.

SEND
specifies that all of the options have been entered and that the message is to be routed. SEND is the final option and must be followed by a space or an end-of-data.

TIME=value
"value" is the time at which you want the message to be delivered. You can specify the time in one of the following four ways:

hhmm
where "hhmm" is an absolute time in the range 0001–2400. For example, TIME=1145 causes the message to be transmitted at 11.45 am. The minutes value must be less than 60.

+hhmm
where "hhmm" is the number of hours and minutes from the current time. The minutes value must be less than 60. For example, TIME=+0720 means that the message is to be transmitted in 7 hours and 20 minutes from now. A value of TIME=+2400 means the same as DATE=+1.

+mm where "mm" is the number of minutes from the current time. This value must be in the range 0–99. So, for example, a value of TIME=+75 causes the message to be transmitted 1 hour and 15 minutes from now. The values TIME=+90 and TIME=+0130 both cause the message to be transmitted in 90 minutes time.

+m where "m" is the number of minutes from the current time. This value must be in the range 0–9. So, for example, a value of TIME=+5 causes the message to be transmitted 5 minutes from now.

If you specify a delivery time on the current day that falls within the past hour, it is interpreted as a request for immediate delivery. An earlier time than that is considered already passed and is treated as an error. The following message is issued:

```
TIME ALREADY PASSED
```

Note that, if the current time is 00.15, T=2345 is interpreted as 23.45 today because there has been a change of date. The message is not therefore transmitted immediately.

Logical unit destinations

This section describes the use of the CMSG transaction to send messages to logical units. For details of message handling within CICS subsystems, and of the use of the message-switching transaction at subsystem terminals, see the appropriate CICS/OS/VS subsystem guide.

Each logical unit in a CICS-SNA network is identified by a single terminal identifier and, if the logical unit is capable of receiving message text, messages may be routed to it in the same way as they are routed to non-SNA terminals. Routing by operator identifier may also be employed if the logical unit supports operator signon, and SNA and non-SNA destinations may be specified in the same ROUTE option.

The destination for a message sent to a logical unit can be a display or printer device, or it can be a data set or an application program in a subsystem controller. To the message sender, the destination behaves like a “terminal”, and any necessary formatting is performed by the CMSG transaction or within the subsystem controller itself.

Logical device codes

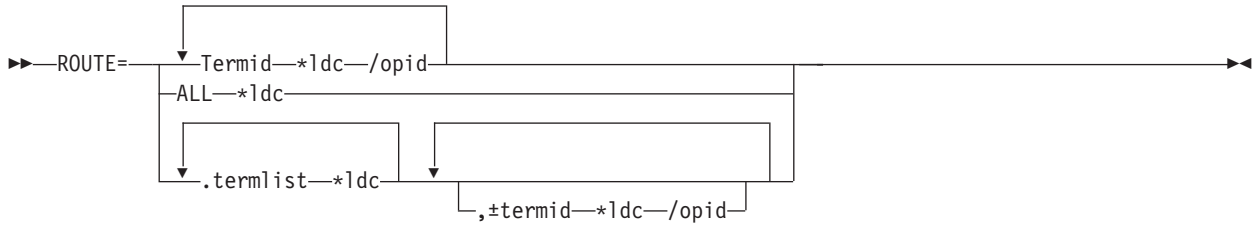
Some types of logical unit (LU) can be used to get access to more than one resource within a subsystem. For example, data sent to a 3601 LU may be intended for an IBM 3604 Keyboard Display, an IBM 3618 Administrative Line Printer, or some other element of the IBM 3600 Finance Communication System. The facility provided by CICS to permit destination selection within LUs of this type is the logical device code (LDC).

The LUs that support destination selection by LDC are:

- 3601 LU
- 3770 batch LU
- 3770 batch data interchange LU
- 3790 batch data interchange LU.

For the user of the message-switching transaction, the LDC is a 2-character mnemonic code whose meaning is defined by the CICS installation. It may be used to qualify an LU destination by including it in the ROUTE option in the syntax diagram that follows:

ROUTE



where:

***ldc**
is a 2-character LDC mnemonic preceded by an asterisk (*) that qualifies the destinations. The *ldc parameter may qualify an LU identifier (termid), a general broadcast (ALL), or a terminal list table specification (.termlist). The *ldc parameter applies only to LUs, not to any start-stop or BSC terminals.

Different LDC mnemonics may be included in one ROUTE option specification; however, all destinations for one message must indicate the same device type.

Termid*ldc
associates an LDC mnemonic with an LU identification.

ALL*ldc
is a general broadcast to all terminals with the same LDC mnemonic qualifying all LUs.

Termlist*ldc
.termlist*ldc
qualifies all entries in this terminal list table with the specified LDC mnemonic. This overrides any LDC specified within the terminal list table. This LDC specification does not apply to start-stop or BSC terminals.

Note: This qualification of a TLT occurs before any succeeding TLTs or +/- entries are processed, see example 9 in “Examples of 3600 and 3770 batch destinations” on page 526.

+termid*ldc/opid
adds a destination, if not a complete duplicate to any contained in the requested TLTs.

-termid*ldc/opid
deletes duplicate destinations resulting from the requested TLTs. A -termid*ldc, without an opid, deletes all destinations of that termid*ldc (with or without operator identifiers) resulting from the requested TLTs. A -termid/opid, without an LDC mnemonic, deletes all destinations of that termid/opid (with or without LDC mnemonics) resulting from the requested TLTs.

If a destination is specified by /opid without termid, it becomes termid/opid, in which termid is the identifier of the first terminal or LU to which an operator with that identifier is currently signed on. If no such terminal or LU is found, the destination is not valid and the operator is notified.

Examples of 3600 and 3770 batch destinations

Put your short description here; used for first paragraph and abstract.

Purpose

This little command copies things.

DITA

Parameters

from *this*
copy from somewhere

to *that*
to somewhere else

Sample

This little sample copies “here” to “there”:

```
copy from here to there
```

Message-switching responses

Every input from the terminal receives a response. Most responses are preceded by indicators where the letters before OK represent the first character of each of the CMSG options (except CANCEL) as follows:

```
D    DATE
E    ERRTERM
H    HEADING
I    ID
M    MSG
O    OPCLASS
P    PROTECT
R    ROUTE
S    SEND
T    TIME
```

These indicators identify the options that have been processed and that are currently in effect.

Errors may occur because of:

- Syntax (for example, misspelled option, unbalanced parentheses, terminal identifier more than 4 characters, invalid option separator, and message and destination not provided).
- Specification (for example, the specified terminal has not been defined to CICS).
- Operation (for example, operator not currently signed on to the system).

Syntax errors within an option cause it to be rejected by the message-switching routine. To correct a known error, reenter the option before typing the SEND keyword.

Syntax errors cause no data to be processed in that input beyond the detected error. All following data must be reentered to be included.

A specification or operation error message may be written to the terminal with an indicator that shows that the option in question has been processed and is still in effect. Two possible error messages are:

1. M OK TERMID GT 4 CHAR AT T004A

   This message has been issued because the terminal identifier is greater than 4 characters. The absence of the R indicator preceding the message shows that ROUTE is not currently in effect and that the message must be reentered.

2. MRHS OK INVLD TERMINL ID T004

   This message has been issued because the specified terminal is not known to CICS. The R indicator shows that ROUTE is still in effect. Operator action as a result of this type of message is either:

   - A space (entry deleted)
   - Termid/opid specifying a termid, or operator identifier, or both (entry changed)
   - Any option (the route option or another option may be entered).

   Note: If an option is entered, the error message is repeated unless a new ROUTE option is entered before SEND. If the incorrect entry is changed or deleted by responding with a space or termid/opid, the message is routed unless other errors are still present. These errors are indicated by another message. If an option is entered, it signifies normal entry mode in which multiple options can be entered requiring another SEND option to end the input.

The following is a list of all possible responses from the message-switching program (DFHMSP). Because of device-dependent considerations, all responses are limited to 25 characters plus the indicators.

ALL ROUTE ENTRIES DELETED
A space entered in reply to a incorrect destination message caused the remaining destinations to be deleted, leaving no destinations to which to route the message. This condition forces a return to normal option mode, and a ROUTE option must be reentered to provide the message destinations.

CANCELED - TEMP STRG ERR N
N has a value in the range 1–4.

- N=1. An error occurred on a temporary storage (TS) request.
- N=2. An error occurred on a temporary storage (TS) request. A possible cause of this response, not related to a TS problem, is a request for a TS record that does not exist. This can happen if:
  - The transaction identifier is entered followed by a C. This extended transaction identifier (CMSGC) normally requests a continuation of a message-switching transaction, but no previous status information exists for this terminal. See “Message-switching continuation” on page 513.

    Note: A CICS cold start of temporary storage causes the loss of any status saved from a previous CICS process.
  - The transaction identifier is entered in lowercase on a 3270 family display device that is defined as a dual case keyboard, and for which no previous message-switching transaction status exists.

    Note: The transaction identifier in the terminal input/output area (TIOA) is not translated. To correct this problem, enter the transaction identifier in uppercase.

N=2 also occurs when transaction routing is being used and indicates that the transaction identifier used to initiate the task is different to that specified in the CICS system definition.

- N=3. An error (most likely caused by TS) occurred on a DFHBMS TYPE=TEXTBLD request.
- N=4. An error (most likely caused by TS) occurred on a DFHBMS TYPE=PAGEOUT request.

CLEAR/CANCL or cont input|route|msg
An operator on a 3270 family display device pressed the CLEAR key and the message states continue (CONT):

INPUT
The last option in the previous input is complete. Enter an option to continue.

ROUTE
The last option in the previous input is ROUTE and a closing parenthesis cannot be found. Continue entering ROUTE destinations.

MSG
The last option in the previous input is MSG, and a single quotation mark to end the message cannot be found, or a single quotation mark is found as the last character of the input, and might be the first of a pair of single quotation marks. If the previous input does not end with a single quotation mark, continue entering the message. If the previous input ends with a single quotation mark, enter another single quotation mark to continue the message (which causes a single quotation mark to be included), or enter any option that causes the message to be considered complete.

Note: If the operator replies by again pressing the CLEAR key (two consecutive times) it has the same effect as entering CANCEL for the last 6 characters of the input.

CONTINUE INPUT
A SEND option has not been processed, and a ROUTE or MSG option was not being continued. Continue by entering any option.

CONTINUE INPUT OR MSG
The previous input ended with a single quotation mark with the MSG option being processed. This single quotation mark could specify the end of the MSG option or could be the first character of a pair of single quotation marks (signifying that a single quotation mark was to be included within the message).

If the next input starts with a single quotation mark, the message is continued with a single quotation mark included in the message. A nonpaired single quotation mark is required to end the MSG option.

If the next input starts with any character other than a single quotation mark, it is treated as the beginning of a new option and the MSG option is considered complete.

CONTINUE MSG
The previous input ended with the MSG option being processed, but no single quotation mark ended the entered text. Continue the message.

CONTINUE ROUTE OPERAND
The ROUTE option was being processed in the previous input and a space or an EOD was found. Continue the ROUTE option.

DATE ALREADY PASSED
A delivery date before the current day is invalid.

DATE INVALID AT xxxxxxxx
DATE must be in one of the following forms:

```
yy.ddd      year (00–99) and day (001–366)
mm/dd       month (01–12) and day (01–31)
mm/dd/yy    month (01–12), day (01–31),
                           and year (00–99)
D=+d        day (0–4)
```

DATE NOT xx/xx at xxxxx
DATE must start with 2 digits followed by a slash (for month), or a period (for year). This message is issued if the third character of DATE is not a slash or a period.

Note:

1. September 5, for example, must be specified as 09/05 and not as 9/5.
2. The date must be valid, not 06/31 for example.

DATE TOO FAR IN FUTURE
Future delivery may not be specified after 0359 hours on the fourth day from the current day. The error is either:

1. D=+n, where n is greater than 4, or
2. DATE specified explicitly as being after the fourth day from the current day, or
3. A condition of DATE and TIME that exceeds the stated limit.

If conditions 1 or 2 occur, the DATE option is not in effect (the D status indicator is not present). If condition 3 occurs, both the DATE and TIME options are currently in effect (the D and T status indicators precede the message), and one or both of them must be changed to a valid delivery time or the error occurs again.

ERRTERM INVALID AT xxxxx
An ERRTERM must be 1–4 characters.

ERRTERM INVLD/UNSUPP xxxx
A terminal identifier specified for the ERRTERM option does not have an entry specified in the CICS terminal definition, or it is for a terminal type not supported by BMS, and is therefore invalid. Valid replies by the operator to this are:

1. A terminal identifier or ORIG in the same format as a normal ERRTERM destination (without entering the option ERRTERM or E). The message is routed following this input unless the new ERRTERM specified is invalid. In this case, the message is repeated.
2. A space as the first character (followed by an EOB or comments). This causes the entire ERRTERM specification to be eliminated. The message is routed following this input.
3. Any option that causes a return to normal option processing mode. If another ERRTERM option is not entered to correct this condition, the same error message is repeated after SEND has been reentered.

HEADNG NOT YES/NO AT xxxx
A HEADING option may only be specified as follows:

```
HEADING
HEADING=YES
HEADING=NO
H
H=YES
H=NO
```

ID OPERAND GT 62 CHARS
An ID option must be from 1–62 characters in length, and enclosed within parentheses.

ID OPERAND MISSING
The ID option requires the identifier to be enclosed within parentheses.

ID PARAM MISSING OR NO)
The ID option must be from 1–62 characters in length and enclosed within parentheses.

INVALID LDC AT xxxxxxxx
The LDC mnemonic is not valid for the logical unit, or a previous entry contained an LDC mnemonic that indicated a different device type.

Note: A missing LDC mnemonic can cause this message if the default mnemonic for that station indicates a different device type than a previously indicated mnemonic (specified or defaulted).

INVALID OPND AT xxxxx
An option xxxxx is invalid (possibly caused by a keying error). No data beyond this point was processed. The status indicators represent the options that are in effect.

INVALID TBL SFX AT xxxxx
A terminal list table suffix must be 1 or 2 characters preceded by a period.

INVLD OPERATR ID AT xxxxx
An operator identifier must be 1, 2, or 3 characters preceded by a slash.

INVLD SEPARATOR AT xxxxx
The end of an option must be followed by a space, a comma (except for SEND), a carriage return (new-line character), or end of data (EOD).

INVLD TERMINL ID xxxx
There is no entry in the CICS terminal definition for the terminal identifier specified for a ROUTE destination. The valid operator reply is one of the following:

1. A terminal identifier, or operator identifier, or both, in the same format as a normal termid/opid ROUTE destination (without entering the option ROUTE= or R=) that replaces the invalid destination.
2. A space as the first character, which causes the invalid destination to be deleted.
3. Any option that causes a return to normal option processing mode. If another ROUTE option is not entered, the same condition exists after the SEND option is reentered.

Note: 1 and 2 cause the message to be routed if the resulting destinations are valid; otherwise, another error message is issued.

LDC NOT 2 CHARS AT xxxxx
The LDC mnemonic that follows the * in the ROUTE destination must be 2 characters.

MESSAGE HAS BEEN ROUTED
A SEND option has been processed and the message has been routed to its destinations. This is the normal completion of a message-switching transaction.

MSG NOT SENT - NO RECEIVER
This condition is caused by an OPCLASS option specified without a ROUTE option and no operators, with the specified operator class, or classes, currently signed-on to terminal types supported by BMS. To correct this situation, a new destination must be entered by the ROUTE option, or OPCLASS option, or both.

NEED OPCLASS OR ROUTE
The SEND option was entered without specifying any destinations for the message. To correct this situation, a destination must be entered by the ROUTE option, or OPCLASS option, or both.

NO INPUT - REENTER
A transaction identifier was entered followed by end-of-data (EOD), or the number of backspace characters entered caused the deletion of all the data (including the position following the transaction identification, if present). If the input does not start with the transaction identifier, the terminal remains in conversation. Otherwise, it is not in conversation.

NO MESSAGE PROVIDED
The SEND option was entered without specifying the message. To correct the situation, provide the message.

NO MSG STRT QUOT AT xxxxx
A MSG parameter must be enclosed within single quotation marks. A single quotation mark must follow an M or MESSAGE.

NO TRMID FOR LDC AT xxxxx
The destination started with an asterisk *. A logical unit identification must precede the LDC mnemonic.

NO TERM LIST TBL-xx
An entry for a terminal list table with suffix xx (DFHTLTxx) was not found in the CICS system definition.

OPCLASS NOT 1-24 AT xxxxx
An OPCLASS parameter must be in the range 1–24.

OPERATORS NOT SIGNDON nnn
The ROUTE option specified operator identifications without terminal identifiers (that is, ROUTE=/opid). nnn operator identifications are not currently signed on.

The valid reply to this is to enter one of the following:

1. A space as the first character, which deletes the nnn destinations.
2. The character D (or any other character) which causes each of the nnn destinations to be individually displayed through the message OPRTR ID NOT SIGNDON.
3. Any option that causes a return to normal option processing mode. If no further ROUTE options are entered, this same condition exists if an operator does not sign on before the SEND option is reentered.

Note: This message is only issued if nnn is greater than 5; otherwise, each destination is individually displayed. Reply 1 causes the message to be routed if the resulting destinations are valid; otherwise, another error message is issued.

OPRTR ID NOT SIGNDON /xxx
An operator with identifier xxx, specified as a destination (entered either by the terminal operator or included as an entry in a TLT whose suffix was specified), is not currently signed on to any terminal controlled by CICS.

The valid reply to this is to enter one of the following:

- A terminal identifier, or operator identifier, or both in the same format as a normal termid/opid ROUTE destination (without entering the option ROUTE or R), which replaces this /xxx destination.
- A space as the first character, which causes the /xxx destination to be deleted.
- Any option that causes a return to normal option processing mode. If no further ROUTE options are entered, this same condition exists if an operator with identifier xxx does not sign on by the time the SEND option is reentered.

Note: The first two replies cause the message to be routed if the resulting destinations are valid, otherwise, another error message is issued.

PROTECT OPND NOT 1-2 CHAR
The PREFIX option of the PROTECT option must not specify more than 2 characters.

ROUTE INVALID AT xxxxx
A field separator entered before the first ROUTE parameter, or 2 consecutive field separators were found. Reenter a valid ROUTE option.

SPACE MUST FOLLOW TRANSID
One or more spaces must follow the transaction identifier before the first option can start. (The character C causes a continuation from the last input preceding a normal completion or a cancelation.)

TERMID GT 4 CHAR AT xxxxx
A terminal identifier must be 1–4 characters.

TERMID NOT +/- at xxxxx
All TLT suffixes must be specified before any individual destination (which must be preceded by a + or -) may be entered. A ROUTE option that specifies a TLT suffix may not be continued on a succeeding input. If a ROUTE option includes more than one parameter, it must be enclosed in parentheses.

TERMINATED BY OPERATOR
The operator entered CANCEL as the last 6 characters of the previous input, or pressed the CLEAR key twice in succession on an IBM 3275 (or 3277) display station. The previous input is canceled and the terminal is **not** in conversational mode with CICS message switching.

TIME ALREADY PASSED
A delivery time for the current day, more than one hour before the current time is considered already passed and is invalid.

If the message is for immediate delivery, enter T=+0, to replace the TIME option that is still in effect.

If the message is for delivery at the specified time on a future day, enter the appropriate DATE option. The specified time is still in effect.

If the message is for delivery at a different time, reenter the TIME option.

TIME INVALID AT xxxxx
TIME must be one of the following:

hhmm
0001–2400 (mm<60)

+hhmm
0000–2400 (mm<60)

+mm 00–99

+m 0–9

where hh=hours and mm=minutes.

TIME NOT 4 CHARS AT xxxxx
A TIME parameter must be a 4-digit number if it is not preceded by +.

TOO MANY TBL SFX AT xxxxx
A maximum of 10 TLT suffixes may be specified.

UNBALANCD PARENS AT xxxxx
A ROUTE or OPCLASS option started with a parenthesis but did not have one at the end, or ended with a parenthesis but did not have one at the beginning. xxxxx represents the characters near the relevant end of the option.

UNSUP OPRT TRMNL xxxxxxxx
An operator identifier without a terminal identifier was specified as a ROUTE destination; however that operator is signed on to a terminal that is a type not supported by BMS and is, therefore, an invalid destination.

The valid reply to this is the same as for the message OPRTR ID NOT SIGNDON.

UNSUP TERMINAL xxxxxxxx
A terminal identifier, specified as a ROUTE destination, is for a terminal type not supported by BMS and is, therefore, an invalid destination.

The valid reply to this is the same as for the message INVLD TERMINL ID.

ZERO LENGTH MSG AT ''
A message must have at least one character enclosed in single quotation marks. A single quotation mark to be included as part of a message must be entered as two separate single quotation mark characters. A message must start with a single quotation mark and may be followed immediately by pairs of single quotation marks.

+DATE INVLD 2330 TO 0030
A DATE=+ option is invalid from 30 minutes before to 30 minutes after midnight.

CMSG examples

Here are some typical examples of the CMSG transaction.

In these examples, # represents a new-line character.

1. CMSG R=ALL,'GOOD MORNING#TODAY IS FRIDAY FEB 1',S
This sends a good morning message to all terminals.

2. CMSG 'GOOD MORNING',R=.G1,T=0900,S
This routes a good morning message to be delivered at 0900 hours to all destinations contained in DFHTLTG1. (Locations in Eastern time zone.)

3. CMSG 'GOOD MORNING',R=(.G2,+CHI4),T=1000,S
This message is for delivery at 10.00 hours to terminals in Central time

zone (DFHTLTG2). Also included as a destination is termid CHI4, which was recently added but not yet included in the TLT.

4. CMSG 'SUPERVISORS' 'MEETING IN 5 MINUTES',O=(4,7),S

This routes the message to all supervisors who are assigned operator class 4 or 7 and are currently signed on to CICS.

5. CMSG R=ALL,'NEW PRICE MODEL 402 - $28.70',S

This sends the price of the new model to all terminals.

6. CMSG 'FRED - WHAT IS CURRENT STOCK ON P/N 4837-LP4# BPK',R=BLD3,H,S

This routes a message to BLD3 asking Fred a question and including the requestor's initials (operator identifier). A heading is included so that Fred also knows the originating termid.

7. CMSG 'PNO 4837-LP4, QTY 26?',R=/BPK,H,S

This shows Fred's answer routed to the terminal at which operator identifier BPK is currently signed on.

8. CMSG 'BRID KADDER ARRIVING 3:15 PST. PLEASE MAKE RESERVATION# M. KRN',R=PADC,H,S

This makes a request for a reservation for a Mr Kadder.

9. CMSG 'SHUTDOWN IN 5 MINUTES',R=ALL,T=1655,S

This sends the shutdown message to all terminals.

Note: Examples 1, 2, 3, and 9 are possible candidates for inclusion in a sequential (BSAM) terminal input stream during startup.

Chapter 27. CREA - create REQUESTMODELs for enterprise beans

Use CREA to generate REQUESTMODEL definitions. The transaction can create definitions dynamically, or write them to the CSD, or both. By default, users who are authorized to use CEDA can use CREA.

CREA is not a general tool for CICS resource definition, nor is it a CSD management tool, except in the restricted sense of the particular objects on which it is designed to operate. These are identified in the description of CREA function that follows.

This section contains the following topics:

- "Function:"
- "Using CREA" on page 536
- "CREC" on page 545

Function:

CREA offers you the following functions in turn:

1. displays a list of beans and bean methods within the JAR file related to an installed DJAR,
2. enables the user to associate transaction IDs with those beans and methods,
3. generates list of the REQUESTMODELs, which best match the tranIDs with the beans and methods in the JAR file,
   - Not every association leads to a REQUESTMODEL definition being generated,
   - CREA produces an optimal list of required REQUESTMODEL definitions.
   - Some of the associations lead to the generation of REQUESTMODEL definitions with names that include wildcard characters.
4. presents each REQUESTMODEL in that list to you, with the option to:
   - install the REQUESTMODEL in CICS,
   - define the REQUESTMODEL to the CSD,
   - both install and define, or
   - ignore, (or skip) the REQUESTMODEL.
5. Ends by presenting to you a list of the REQUESTMODEL definitions that you have installed and those that you have defined.

   CREA includes in this list any REQUESTMODELs installed in CICS that are not used by the DJAR (but refer to beans within the DJAR), and also reports on any tranIDs which are used but not currently installed.

Note:

1. CREA only works on installed definitions,
   - You can choose which DJAR definitions are examined by CREA by choosing which definitions are installed at the time that you invoke CREA. In this way you can avoid having CREA present large amounts of information relating to DJARs that do not concern your current activity.
2. Any REQUESTMODELs already installed into CICS are analyzed and tranIDs defined on those REQUESTMODELs are matched to any appropriate beans or methods,

© Copyright IBM Corp. 1982, 2011

Using CREA

CREA is defined in the DFHADST group and is automatically installed. Your CICS system must be set up to run Language Environment conforming C applications and Java.

CREA presents you with a series of screens as it progresses through its functions.

Note: Choose a suitable code page to connect to your CICS system. If you don't, you'll see the square bracket characters **[** and **]** displaying strangely. One suitable code page is 1047 (United States English).

Starting CREA

1. Start the transaction by entering `CREA` (optionally followed by the name of a DJAR definition), on the command line of the screen.
2. Press ENTER.

If the DJAR name supplied is valid (that is: if it exists in CICS and is in a resolved state), you are shown the TranID Specification screen (Figure 208 on page 538), which presents the functions available within CREA.

If you do not give the name of a DJAR definition, or if the DJAR definition you name is invalid, you are shown the initial screen (Figure 207 on page 537) and a message indicating the state of the DJAR, or indicating that the DJAR was not found, (as is appropriate).

Command syntax

CREA

Command options

djarname
specifies the identifier of a DJAR definition that you want to work with. The names of the beans and methods in the JAR file associated with this DJAR definition are displayed alongside the tranIDs associated with each.

Response to CREA command

```
   CREA   REQUESTMODEL Definition Transaction           APPLID

   Please enter the name of a DJAR you want to work with.

   DJAR name: ________


   If you want to restrict the display of beans and methods within the
   DJAR to those associated with a particular transaction then you can
   use the tranID filter.  The * character can be used as a wildcard.


   TranID filter: ____



<Message Line >
 PF1=Help    3=Quit
```

Figure 207. CREA transaction: initial screen

If you enter CREA with a valid DJAR name as a parameter, the initial screen (Figure 207) is not presented to you. Instead you are taken directly to the TranID Specification screen (Figure 208 on page 538).

If you enter CREA with an invalid DJAR name as a parameter, the name you entered appears in the DJAR name field of the initial screen Figure 207.

If the status of the named DJAR definition is **resolving**, you have the opportunity to continue. To do so, press enter again and, if the DJAR is now resolved, you are taken to the TranID Specification screen (Figure 208 on page 538).

This initial screen also offers you the opportunity to enter a tranID with which to filter the beans and methods displayed. The filter can be blank to display all beans and methods, or contain a string with optional wildcard characters ('*'). If the filter field is not blank, only those beans and methods that match the filter are displayed.

TranID Specification

When a valid DJAR name has been entered, the TranID Specification screen is displayed. It shows tranIDs alongside the bean and method names that are defined in the deployment descriptor of the JAR file associated with the chosen DJAR definition.

```
CREA  REQUESTMODEL Definition Transaction           APPLID

Associate a transaction ID with a bean or bean method.

DJAR name: SampleEJB                          TranID filter: ____

TranID  Bean/Method/Parameters


CIRP     CICSSample
CIRP >      getCustomerInfo
               int
CIRP >      getEJBHome (inherited from EJBObject)
CIRP >      getHandle (inherited from EJBObject)
CIRP >      getPrimaryKey (inherited from EJBObject)
CIRP >      isIdentical (inherited from EJBObject)
               javax.ejb.EJBObject
CIRP >      remove (inherited from EJBObject)



<Message Line>
 PF1=Help  3=Quit    5=Create ReqMods                      9=Hide Methods
```

Figure 208. CREA transaction: TranID Specification screen

The example given in Figure 208 shows the methods that were found in the Deployment Descriptor of the JAR file associated with DJAR definition *SampleEJB* .

The tranID that is to be used to execute requests (to be identified at run time by an installed REQUESTMODEL resource definition) precedes each of the beans and methods. If there are any currently installed REQUESTMODELs that relate tranIDs to any of the beans or methods in the JAR file, the REQUESTMODEL tranID is shown next to the relevant bean or method. In this example, it is the default value CIRP in every case.

The way that the tranID is determined is this:

- If the *method* already has a tranID relationship, either supplied by you during a previous use of CREA, or defined by an installed REQUESTMODEL, it is displayed, otherwise,
- If a *bean tranID* is known, or if you supply one, any methods on that bean default to the bean tranID.
- For all remaining beans and methods, the Tranid of the default REQUSTMODEL is shown.
- If no default is defined, a default tranID of CIRP is shown.

The rules that influence the use, or not, of default values are :

- After you have entered a tranID, or it has been defined by an installed REQUESTMODEL, it is not replaced by a default value.
- However, if you change a *bean* tranID, all *method* tranIDs for that bean, that have taken a default value, change to match the bean tranID.
- If you change a tranID to spaces, the tranID takes the default value of the bean tranID, and subsequently reacts to any relevant changes to the default, as appropriate.
- If you change the tranID of a method to match that of a bean, then the method tranID ceases to be an absolute value, but becomes one which takes the default of the bean tranID. It subsequently reacts to any changes to that default.

You can see which tranIDs take default values if the bean tranID changes, because:

- The tranIDs that are explicitly defined for beans or methods (that is, they relate to beans or methods because of an installed REQUESTMODEL, or they were entered by you) are referred to as defined tranIDs. They are displayed in white.
- TranIDs that take default values (either CIRP, the overriding default REQUESTMODEL tranID, or the bean tranID defaulting for each of the bean's methods) are displayed in green and are further highlighted with a **>** character two characters to the right of the tranID entry field.

When entering tranID values:

- an **'** sign means - use the value from the tranID field for the bean or method on the line above, and if that is also an **'** sign, use the value above that, and so on.

  An **'** sign is not valid in the first tranID field displayed. If one is used there, and in any consecutive fields, then those tranIDs are not changed from their current setting.
- a **** sign in the tranID field of any line, causes the display to scroll, making this line the top line on the screen. Only the first **** from the top of the present display is honoured.The tranID field into which the **** is entered is unchanged and is refreshed with its original contents.

At anytime during the process of assigning tranIDs, you can change the tranID filter. The filter causes the display to list only those beans and methods that match the filter. The filter can be blank to show all methods, or can contain a string with optional wildcard characters ('*'). Changes to the filter do not cause associations between tranIDs and beans and methods to be lost.

If you enter a new DJAR definition name, a message is displayed, indicating that you have changed the DJAR name, and prompting you to press enter to continue. If any other key is pressed as a response to this message, the DJAR definition name reverts to the current name.

- This action is equivalent to restarting CREA after quitting.
- It allows you to change the DJAR definition name if you realize that the DJAR definition that you have started to work with is not the one you intended.
- Changing the DJAR name results in the loss of all of the changes you have made to tranID & bean/method associations since this screen was first displayed.
- The screen is redisplayed, showing the new DJAR definition's beans and methods, along with any known tranIDs.
- A **+** symbol is displayed in the line above the top method, or the line below the bottom method if more data is available.
- You can scroll up and down the contents of the JAR file using the **PF7** and **PF8** keys, and you can hide the methods associated with each bean by pressing **PF9**. When **PF9** is pressed to hide the methods, the **PF9** legend will change to read **Show Methods**, and this is reversed when it is pressed again.
- Keys legends **PF7**, **PF8** and **PF9** only appear when they are relevant. If all the information appears on one screen, **PF7** and **PF8** are not displayed. At the top or bottom of the information, only one of them is shown. If there are no methods associated with the beans, being displayed, **PF9** is not shown.
- When you scroll down, the line displayed at the bottom of the previous screen, becomes the top line of the next screen, and when you scroll up, the previous top line becomes the bottom line of the next screen.
- Pressing a function key or enter commits any changes entered on the current displayed window.
- The clear key cancels all changes made to the current screen displayed window.

When you have finished associating tranIDs with beans and methods, you can press **PF5** to create the REQUESTMODEL definitions reflecting the tranID & bean/method relationships. This takes you to the REQUESTMODEL Specification screen (Figure 210 on page 541).

Setting tranIDs for different kinds of remote method calls

You might consider three kinds of remote method calls that may be invoked for a given enterprise bean.

- The business methods, as defined in the component interface class
- The infrastructure methods, inherited from the component interface super class (javax.ejb.EJBObject)
- The methods on the home interface, (these are, nonetheless,remotely callable methods)

The primary purpose of CREA is to define REQUESTMODELs for the business methods, but CREA treats all three alike, so if, for example, you decide that a particular method on a bean is to run under a different transaction id from the others, or that all the remote methods for a given bean are to run under the same (but not the default) default transaction id, you can use CREA to arrange this.

In Figure 209 on page 541, a DJAR called HelloWorldEJB, which contains a single enterprise bean called 'HelloWorld', is being processed. This bean has one business method, called 'hello', defined on the component interface. An existing REQUESTMODEL has been defined in the region which maps the 'hello' method of this bean to the 'WIBB' transaction id. All other methods run under the default CIRP transaction id.

Note: WIBB and the CIRP above it appear in white, the CIRPs below WIBB appear in green.

The line which contains the bean name ('HelloWorld') can be used to set a default transaction id under which all methods on this bean will be invoked (including methods on the home interface). Below this are listed in alphabetical order the business methods of the bean. In this case, there is only one business method. Below the business methods are listed the five methods inherited from the component interface's super class (javax.ejb.EJBObject) - these same methods will be listed for all enterprise beans. Methods on the Home interface are not listed. If you want all the remote methods, including those on the home interface, to run under a specific transaction id then supply new default against the line which lists the bean name. This causes a REQUESTMODEL to be created with an interface type of 'both'. All other REQUESTMODELs are generated with an interface type of ‘remote’ and are only applied to the component interface.

If multiple overloaded method names (methods with the same name but different parameters) exist for a particular bean then a method name that includes wildcard characters will also be shown amongst the list of business methods. This is not illustrated in Figure 209 on page 541. You can set a transaction id against this method name (that includes wildcard characters) in order to set the preferred transaction id for all the overloaded instances of that method.

In all cases, the most specific REQUESTMODEL defined is the one honoured at run time. So, a REQUESTMODEL set against a specific method will always be used in preference over one set against a set of overloaded methods, and this overloaded REQUESTMODEL will be used in preference to the default set against the entire bean.

```
CREA  REQUESTMODEL Definition Transaction           APPLID

Associate a transaction ID with a bean or bean method.

DJAR name: HelloWorldEJB                        TranID filter: ____

TranID  Bean/Method/Parameters


CIRP     HelloWorld
WIBB        hello
                java.lang.string
CIRP >      getEJBHome (inherited from EJBObject)
CIRP >      getHandle (inherited from EJBObject)
CIRP >      getPrimaryKey (inherited from EJBObject)
CIRP >      isIdentical (inherited from EJBObject)
               javax.ejb.EJBObject
CIRP >      remove (inherited from EJBObject)



<Message Line>
 PF1=Help  3=Quit    5=Create ReqMods                      9=Hide Methods
```

Figure 209. CREA transaction: TranID Specification screen

REQUESTMODEL Specification

The REQUESTMODEL Specification screen (Figure 210) shows you, one by one, each of the REQUESTMODEL definitions that is to be created.

```
CREA CICS REQUESTMODEL Definition Transaction           APPLID

Create a REQUESTMODEL definition

Group: _____   Define to CSD: N   Install to CICS: N   Replace Dups: N

REQUESTMODEL name CREA_____              CORBASERVER EJB1

TranID CIRP

Bean CICSSample_____________________________________________
_____________________________________________________________
_____________________________________________________________
_____________________________________________________________

Operation *_________________________________________________
_____________________________________________________________
_____________________________________________________________
_____________________________________________________________


<Message Line>
 PF1=Help   3=Quit     8=Skip ReqMod     ENTER=Create ReqMod
```

Figure 210. CREA transaction: REQUESTMODEL Specification screen

For each REQUESTMODEL definition, the following information and input fields are shown shown:

Group: _____
If you select **Define to CSD** by entering a **Y**, .The **Group** field must contain the

name of the group that is to hold the definition. The value given for **Group** becomes the default for each subsequent REQUESTMODEL definition, but can be changed again at any point.

Define to CSD: _
Indicate where you wish to create the REQUESTMODEL definition by entering a **Y** in the **Define to CSD:** field, or the **Install to CICS:** field, or both.

If you select **Define to CSD:**, the **Group** field must contain the name of the group that is to hold the definition. The value given becomes the default for each subsequent REQUESTMODEL definition, but can be changed again at any point.

Install to CICS: _
Indicate where you wish to create the REQUESTMODEL definition by entering a **Y** in the **Install to CICS:** field, or the **Define to CSD:** field, or both.

Replace Dups: _
To overwrite existing definitions (either already installed in CICS, or already defined in a group on the CSD), enter a **Y** in the **Replace Dups:** field. If this field is **N**, the existence of duplicate definitions is reported as a warning. When the warning is issued, you can choose to replace this duplicate definition by pressing **PF9**, or skip to the next definition with **PF8**.

Initially a default of **N** is used. When you change this, the change become the new default for subsequent REQUESTMODEL definitions.

REQUESTMODEL name: _____
For each REQUESTMODEL definition, a generated name is shown. As each is presented, you can:

Accept
this name and move on to the next REQUESTMODEL definition. For details about the naming of REQUESTMODEL definitions, see “CREA generated REQUESTMODEL names” on page 543.

To create the definition, press enter. Successful creation of the REQUESTMODEL definition causes the next to be displayed.

Also displayed is a message indicating the name of the REQUESTMODEL definition that you have just completed and where it was defined (installed into CICS, written to the CSD, or both).

When the last REQUESTMODEL definition has been processed, the Results screen is displayed with no message indicating creation (as that information is available as part of the Results information).

If the creation of a REQUESTMODEL definition fails for any reason, a message is returned to the user, with a redisplay of the screen that requested the creation of the failing REQUESTMODEL definition. This gives you the opportunity to correct any problems with the group or the name.

When you ask for both **Define to CSD:** and **Install to CICS:**, the creation of the REQUESTMODEL definition into CICS is attempted first. If the creation does not work, the definition is not written to the CSD. If the creation of the REQUESTMODEL definition into CICS works, but the definition to the CSD fails, and you make changes to the name or tranID, the REQUESTMODEL installed in CICS is altered (or deleted and recreated) as appropriate.

Override
the generated name, then accept the new name and move on. For

details about the naming of REQUESTMODEL definitions, see “CREA generated REQUESTMODEL names.”

To create the definition, press enter. Successful creation of the REQUESTMODEL definition causes the next to be displayed. The remainder of **Override** processing is the same as for **Accept**.

Skip this REQUESTMODEL definition and move to the next.

If you do not want to create the REQUESTMODEL definition displayed, press **PF8** to move to the next definition, or to the Results screen if there are no more. If you choose to skip the REQUESTMODEL definition after an error has occurred, and the REQUESTMODEL definition was installed into CICS, but could not be defined to the CSD, the REQUESTMODEL definition is removed.

Quit at any point.

To quit, press **PF3**

CORBASERVER: ____
The CORBASERVER, field is not available for change.

TranID: ____
The TranID field is not available for change.

Bean: ____
The Bean field is not available for change.

Operation: ____
The Operation field is not available for change.

CREA generated REQUESTMODEL names

CREA generates and displays a name for each REQUESTMODEL definition that is required. That generated name is produced as follows:

- The initial name is generated by suffixing a count to **CREA**, producing *CREA1*, *CREA2*, and so on, to *CREA9999*.
- When the count becomes large enough that the prefix and suffix together exceed eight characters, the prefix is truncated to allow the process to continue, so *CREA9999* is followed by *CRE10000* thru *CRE99999*, then *CR100000* thru *CR999999* and so on. If , following this rule, the name reaches *99999999*, then incrementing ceases, and you are presented with the same REQUESTMODEL name again.
- You can overtype this name with one of your choice.
- If you do, you change the basis for the next REQUESTMODEL default name. If your overtyped string includes a trailing count, this is used as a starting point, with 1 being added to this count for each new REQUESTMODEL.

  For example, if you overtype *CREA1* with:
  - *SHEEP*, the following REQUESTMODEL name displayed is *SHEEP1*,
  - *RHINO26*, the following REQUESTMODEL name displayed is *RHINO27*,
  - *GIRAFFE9*, the following REQUESTMODEL name displayed is *GIRAFF10*.

  The same rules about truncation of the prefix apply when the succession of names is based on a string that you previously supplied by overtyping. This is true both when incrementing is on the point of taking you beyond the eight character maximum, and when you supply an over-long prefix by overtyping, for example, if you overtype with *ELEPHANT*, the following REQUESTMODEL name displayed would be *ELEPHAN1*.

Results

When you have finished creating REQUESTMODEL definitions, the REQUESTMODEL results screen Figure 211 is displayed, summarizing what you have done. It lists the REQUESTMODEL definitions that were installed into CICS and the REQUESTMODEL definitions that were written to the CSD. If more than one group was used to hold definitions, each group is listed with the REQUESTMODEL definitions it contains.

```
   CREA CICS REQUESTMODEL Definition Transaction            APPLID

REQUESTMODEL definition is complete.

REQUESTMODELs were processed for DJAR SampleEJB

The following REQUESTMODELs were installed into CICS:
RM1, RM2, RM3, RM4, RM5

The following REQUESTMODELs were written to the CSD in group YOURNAME:
RM2, RM5

The following REQUESTMODELs match one or more beans in the DJAR, but either
do not match any methods on that bean, or have been superceded by a newly
created REQUESTMODEL.  These REQUESTMODELs are no longer needed and may
be deleted:
RM7, RM8

The following REQUESTMODELs contain a bean name that matches one or more of
the beans in the DJAR because wildcard characters were used in the bean name.
These REQUESTMODELs are no longer needed to match the beans in this DJAR and
may be deleted if other beans in other DJARs no longer require them:
RM10, RM11

The following TRANIDs were used, but are not currently installed:
TRN1, TRN2, TRN3, TRN4

<Message Line>
 PF1=Help   3=Quit                          12=New DJAR
```

Figure 211. CREA transaction: REQUESTMODEL results screen

The following messages appear on this screen:

- **REQUESTMODEL definition is complete.** always appears first.
- **REQUESTMODELs were processed for DJAR** *SampleEJB***.** always appears next. The DJAR name shown is the one you specified at the start of this use of CREA.
- The next message is either **The following REQUESTMODELs were installed into CICS:** *list of REQUESTMODELs*, or **No REQUESTMODELs were installed into CICS.**
- then, **The following REQUESTMODELs were written to the CSD in group** *Yourname***:***list of REQUESTMODELs*, or **No REQUESTMODELs were written to the CSD in any groups.** The group name, shown here as *yourname* is whatever you supplied previously, on the REQUESTMODEL Specification panel.
- The following messages *only* appear if appropriate:
  - The following REQUESTMODELs were found to be related to a bean in the selected DJAR, but not matched to a bean or method: RM7, RM8

    Such REQUESTMODEL definitions may be redundant, and could potentially be removed from the CICS system (and from the CSD if applicable).
  - The following REQUESTMODELs were found to be related to a bean in the selected DJAR, but are no longer use by the selected DJAR: RM10, RM11

Such REQUESTMODEL definitions are no longer required by the bean, and could potentially be removed from the CICS system (and from the CSD if applicable).
  - The following TRANIDs were used, but are not currently installed: TRN1, TRN2, TRN3, TRN4

  In these messages, RM7, RM8, RM10, RM11 and TRN1, TRN2, TRN3, TRN4 are examples of names for REQUESTMODELs and TRANIDs. They have no particular significance.
- For each of the three messages above, there is no message to indicate the absence of that condition.
- If the information extends beyond one screen, use **PF7** and **PF8** to scroll up and down the information presented.

To work with another DJAR definition, restart CREA by pressing F12 to return to the initial, (or DJAR Specification), screen (Figure 207 on page 537).

CREC

The CREC transaction, allows you to use those functions of CREA, which display information, but you cannot update either the CSD file or the running CICS system. By default, users who are authorized to use CEDC can use CREC.

Using CREC

Use CREC in the same way that you use CREA. Because CREC only displays information, and changes nothing there are two significant differences to note:

- The TranID Specification screen (Figure 208 on page 538) displays the tranID fields next to the beans and methods in a read only state. The F5 option is unavailable and does not appear.
- The REQUESTMODEL Specification screen (Figure 210 on page 541), and the REQUESTMODEL results screen (Figure 211 on page 544) are not available.

Chapter 28. CRTE - remote transactions

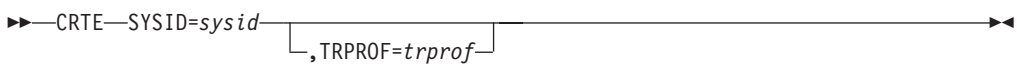
Use CRTE to run transactions that reside on a remote system.

Transaction routing in a multiregion operation (MRO) environment or for intersystem communication (ISC) is usually achieved by making entries in the PCT. The entries define the transaction as remote, and name the system on which it is to be processed. Entering the transaction code then causes CICS to route the requests to the named system.

The CRTE transaction provides you with another way of running a transaction on a remote system. Use CRTE, rather than making special PCT entries, for infrequently-used transactions or when using transactions such as CEMT which reside on all systems.

You must use CRTE when EDF is used to test a transaction running on a remote system. However, because the session has to be established and canceled explicitly, additional signon operations may be required. CRTE can be invoked from a 3270 terminal, but not on LU6.1 links.

CRTE



where:

sysid
: is the name of the system that is to process the transaction. This *sysid* is the same as the 4-character CONNECTION specified, either by using RDO (resource definition online), or the 4-character SYSIDNT specified in the DFHTCT TYPE=SYSTEM definition of the remote system.

trprof
: is the 8-character name of the profile that is used for the session that carries the intersystem flows, if that session uses VTAM. If VTAM is used and this parameter is not entered, the default profile DFHCICSS is used.

The routing transaction verifies that the specified remote system is known and is available. If it is, a message is displayed confirming that a routing session to the required system has been started.

When you have received this message, you can clear the screen and enter the transaction identifier for the transaction that is to be run on the remote system. In fact, you can use the terminal as if it were connected directly to the remote system, although you might be asked to sign on to the remote system before being allowed to run any transactions. The transactions that you can invoke include pseudoconversational transactions, BMS paging transactions, and even CRTE itself. However, you cannot use transactions CEOS and CEOT with CRTE, and you cannot use PA or PF keys to invoke transactions under CRTE.

You end a routing session by entering CANCEL. If you have signed on to the remote system, the CANCEL request automatically signs you off.

When a routing session ends, you get the message:

© Copyright IBM Corp. 1982, 2011

ROUTING SESSION TO SYSTEM *system-id* HAS BEEN TERMINATED

Note:

1. If you use CRTE to route transactions through more than one system, then, to end the routing session, you must press CANCEL the same number of times as you entered CRTE.
2. While a terminal is in a routing session with another system (that is, during the period between entering CRTE and CANCEL) the terminal cannot receive messages delivered by automatic transaction initiation (ATI).
3. The message cannot be delivered immediately if a transaction:
   - Runs on a remote system using CRTE
   - Builds a message using BMS paging
   - Sends the message using a **BMS SEND PAGE** command
   - Does not specify either RETAIN or RELEASE.

   It is delivered when the routing session ends, but only if the terminal was in TRANSCEIVE status when the routing session commenced. Alternatively, you can retrieve the message (while still routed) using BMS paging commands.
4. There are some restrictions on the use of CRTE with BMS chaining of messages:
   - You cannot use CRTE as a chained transaction without losing the logical message from which CRTE was to be chained.
   - You cannot use chaining within a routing session without purging the logical message currently attached to the terminal.
   - You cannot route transactions back to any system in the chain.

Chapter 29. CSFE - terminal and system test

Use CSFE to diagnose terminal problems and software problems. This transaction is mainly intended to be used by system programmers and IBM field engineers.

Use this transaction to:

- Test your terminal
- Activate the system spooling interface trace
- Activate the terminal builder trace
- Activate any of the following: storage freeze, the storage violation trap, and the global trap/trace program exit routine.

For each of these activities, successful completion of the CSFE transaction gives you the following message:

```
DFHFE3301 Transaction complete
```

For information about the use of this transaction, see the *CICS Problem Determination Guide*.

Using CSFE terminal test

You can use CSFE to test your terminal, but only if it is one of the terminals supported by CICS.

You cannot use CSFE to test the following:

- IBM 2780 Data Transmission terminals
- IBM 3600 Finance Communication System terminals
- IBM 3614 Consumer Transaction Facility terminals
- IBM 3735 Programmable Buffered terminals
- Output-only printers (for example, printers of the IBM 3270 Information Display System).

1. You start the transaction by entering CSFE on the command line of the screen, as follows:

   ```
   CSFE
   ```

2. Press the ENTER key. You get the following display:

   ```
   DFHFE3304  Enter PRINT for character set, END to
   terminate.  All other data will be echoed.
   ```

   END
   When you type END on the command line of the screen, the test ends.

   PRINT
   When you enter PRINT, all characters printable or capable of display at the terminal are sent to your terminal.

   Any other input is returned to your terminal (that is, it is echoed), unless it is of the form of one of the other CSFE requests.

Tracing with CSFE

You can use CSFE to request additional trace and diagnostic facilities for the system spooling interface.

© Copyright IBM Corp. 1982, 2011

CSFE SPOOLFE

OFF
switches off the system spooling interface trace and the diagnostic facility.

ON
specifies that additional trace entries of system type are provided for the system spooling interface.

SPOOLFE
is the system spooling interface trace and diagnostic facility.

Terminal builder trace

You can use the CSFE transaction to activate and deactivate the terminal builder trace.

CSFE ZCQTRACE

,AUTOINSTALL
Switches tracing on for autoinstalled terminals.

,OFF
Switches tracing off for individual and autoinstalled terminals.

=termid
Switches tracing on for the terminal or connection identified by termid.

Using CSFE debug functions

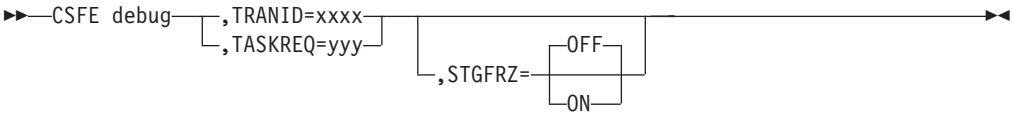
Storage freeze, the storage violation trap and the global trap/trace exit are all debugging functions that are controlled by the CSFE transaction. These functions are described in the following sections.

Storage freeze

You can use CSFE DEBUG to freeze storage that is normally freed during the processing of a transaction. The frozen storage is not freed until the end of the transaction. This freezing of storage can help you in the subsequent diagnosis of problems that might have occurred during processing.

You must code the options in the order shown in the syntax display.

CSFE DEBUG TRANID



Parameters

STGFRZ= {**OFF**|**ON**}
Specifies that you want the storage freeze function to be set on or off. If you omit this option, storage freeze is set to OFF by default. To improve efficiency, storage freeze should be switched off when not required.

TASKREQ= *yyy*
This is an alternative method of identifying the transaction. *yyy* is the PA or PF key definition on a 3270 display device which identifies the transaction for which storage freeze is to apply; for example, TASKREQ=PA1.

TRANID= *xxxx*
Identifies the transaction for which storage freeze is to apply.

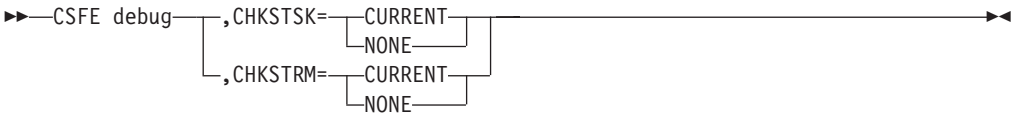
Storage violation trap

Use CSFE DEBUG to activate and deactivate the storage violation trap, which is in the trace program (DFHTRP), the exec interface program (DFHEIP), and the resource manager interface program (DFHERM).

This trap lets you verify the user storage associated with the currently active task. The storage violation trap is entered before the trace program returns to its caller after making a trace entry.

When a storage violation is detected by the trap, an exception trace entry is made, a CICS system dump is produced (if dumping has not been disabled), and the storage violation trap is switched off.

CSFE DEBUG CHKSTSK



Parameters

,CHKSTSK= {**CURRENT**|**NONE**}

CURRENT
Specifies that validity checks are to be performed on the user storage (not the CICS storage) for the current task whenever the trace program is entered. Validity checks are to be performed for the currently active task, as follows:

- Validity checking of the storage chain for the task
- Validity checking of the addresses and storage accounting areas for all storage areas on the transaction storage chain.

NONE Deactivates the user storage checking function.

,CHKSTRM= {**CURRENT**|**NONE**}
The terminal subpools, above and below the line, are to be checked for the CURRENT task or for NONE (no tasks).

Global trap/trace program exit and global kernel exit

You can use CSFE DEBUG with the TRAP or GTRAP operand to activate and deactivate the global trap/trace program exit routine or the global kernel exit routine respectively.

These exits are intended to be used only under the guidance of IBM support personnel.

CSFE DEBUG TRAP

TRAP={**ON**|**OFF**}
The global trap/trace program exit routine is to be activated or deactivated.

GTRAP={**ON**|**OFF**}
The global kernel exit routine is to be activated or deactivated. On each domain call and return and each subroutine call and return, the kernel invokes the global kernel exit routine if it is activated.

Chapter 30. CSPG - page retrieval

Some transactions, especially those performed on screens, produce a response that is too large to be displayed or printed at one time. The whole response is held on direct access storage, and only the first portion (called a page) is displayed. You can manipulate these pages, through basic mapping support (BMS), using the CSPG page retrieval transaction.

You can use paging to view all pages of the response, in any order. All the pages produced by a transaction make up a logical message.

A page retrieval session begins when you enter the first page retrieval command, or when the first page is presented by automatic transaction initiation (ATI). During the session, CICS passes commands directly to BMS terminal paging, without following the normal transaction identification procedure.

When using BMS for IBM 3780 Communication Terminal transactions, switch off the SPACE COMPRESS EXPAND (standard feature), TRNSPCY (special feature), and the INHIBIT IRS (special feature) switches.

Pages to be manipulated with paging commands must have been created by BMS, which must be active in the system to service the commands. Paging commands enable you to:

- Retrieve a page.
- Display information about logical messages queued for your terminal.
- Chain a logical message, that is, retain the previous logical message on direct access storage while you enter a new transaction.
- Purge a logical message, that is, erase it from direct access storage; a paging session must be ended by purging the message.
- Copy a page to another terminal.

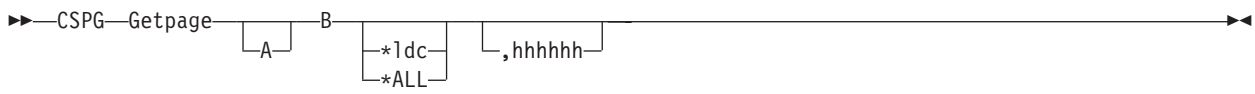
You select the above functions by entering a unique identification code for each function. This code is 1–7 characters long, and is defined by the PGRET, PGCHAIN, PGPURGE, and PGCOPY system initialization parameters.

Do not use codes that are the same as the leading characters of any transaction defined in the PCT.

Retrieve a page

The page retrieval command allows you to retrieve a specified page. If partitions are in use, the page retrieval command refers to the partition in which it has been entered (see “Inquiring on the status of CIND” on page 500).

CSPG GETPAGE



© Copyright IBM Corp. 1982, 2011

Getpage
This is the user-defined identification code for the retrieval transaction, and is 1–7 characters long. This code is defined by the PGRET system initialization parameter. For convenience, make the code as short as possible; for example, P/.

A You can ignore this option unless chaining is being used. It is the level, within a chain of logical messages, of the logical message to be retrieved. You must enclose it in parentheses. It can be one of the following:

n message is at level n

+n message is n levels forward

-n message is n levels backward

P previous level backward

C current level (the default)

N next level forward

L last level.

For more information on levels of messages, see “Message chaining with CSPG” on page 556.

B Defines the next operation to be performed on the logical message at the current level, or at the level specified by the a option It can be one of the following:

n page n

+n page n pages forward

-n page n pages backward

P previous page

C redisplay current page (after clearing the screen)

N next page

L last page

A for an autopaging (that is, hard copy) terminal that is temporarily in paging mode: return the terminal to autopaging mode and output all remaining pages.

This is discussed further under “Using Autopage with CSPG” on page 560.

X enter data that is not recognized as a paging command.

This is discussed further under “Single keystroke retrieval (SKR)” on page 555.

***ldc**
Is the LDC for the device to which you want to send the message. (See “Page retrieval for logical devices” on page 555.)

***ALL**
A special case of *ldc.

Hhhhhh
Is the 6-digit hexadecimal identifier of an undelivered logical message to be retrieved. You would normally determine this identifier by entering a page query command (see “Display information about logical messages” on page 556).

If you do not specify an identifier, the oldest message is retrieved. After a paging session has started, you can omit hhhhhh because all subsequent pages are from the current message.

In the following examples, P/ is the page retrieval transaction identifier:

P/7 Retrieve page 7 of this message.

P/N Retrieve next page (following the one being displayed) of this message.

P/(N)9 Retrieve page 9 of the next message (of several that have been chained).

Single keystroke retrieval (SKR)

If you use a 3270 display device under VTAM, you can use PA or PF keys, as defined in the SKRxxxx system initialization parameter, to enter page retrieval commands, and greatly reduce the number of keystrokes necessary in a page retrieval session.

After the session has ended, the PA and PF keys can again be used for initiating transactions in the usual way. The *CICS System Definition Guide* gives further information on this topic.

Page retrieval for logical devices

Your application program can send messages to a communication system such as the IBM 3600. Your application program can also send different pages to each of the logical devices within the system. These pages must be grouped as a single logical message, held in storage.

Each part of a logical message is accompanied by a logical device code (LDC) that identifies its destination. When retrieving the pages for a logical device, you must specify the LDC for that device. However, you do not need to do this if the user-written program in the logical device controller provides a default LDC mnemonic.

If you specify the page number and *ldc, delivery begins with the specified page for that logical device. If the logical device is in AUTOPAGE status, all pages for that logical device, starting with the requested page, are written. If the logical device is in PAGE status, only the specified page is written.

If you omit *ldc, and the application program in the logical device does not provide a default, BMS selects the first page destined for a logical device that is currently in PAGE status. The application program in the logical unit should provide a default LDC, because pages may be delivered to logical devices other than those you originally specified.

In either case, if your request is the first request for a page from a logical message, all pages (within the logical message) destined for logical devices currently in AUTOPAGE status are delivered after the first or specified page is delivered to a device in PAGE status.

The getpageA command supports an *ldc specification for nonrouted messages created for a 3601 logical unit. The various forms of getpageA are interpreted as follows:

- getpageA*ALL-resets to autopaging and delivers all pages for the logical devices that have autopage specified in the terminal definition, but that were temporarily placed in page status by the CICS application programmer.
- getpageA*ldc-resets the specified logical device to autopage and delivers the pages for that *ldc. The command is similar to getpageA*ALL, except that it operates only for the specified logical device.
- getpageA-resets to autopaging and delivers the remaining pages for the logical device for which pages are currently being displayed.

Page retrieval and partitions

The page retrieval commands, including any PA and PF keys used in single keystroke retrieval, refer to the partition in which they are entered. For example, P/N means 'retrieve the next page for this partition'. The partition in which the last paging command is entered is the *current partition*. A paging command for a partitioned message entered on a nonpartitioned screen (that is, after CLEAR) refers to the current partition.

After a transition from base (unpartitioned) state to partitioned state, or a change of partition set (that is, the initial page display, a page display following CLEAR, or following a change in the level of page chaining), the *current page* is displayed in each partition. The current page is initially the first page for that partition. It is updated by page retrieval commands.

Display information about logical messages

Put your short description here; used for first paragraph and abstract.

Purpose

This little command copies things.

DITA

Parameters

from *this*
copy from somewhere

to *that*
to somewhere else

Sample

This little sample copies "here" to "there":

```
copy from here to there
```

Message chaining with CSPG

Usually, when you have viewed the output pages from a transaction, there is no need to retain the pages. However, you might want to initiate a new transaction and still retain the old pages, so that you can view them again after you have displayed output from the new transaction. This is called *message chaining* and can be done using the following command.

CSPG MESSCHNYYYY

Messchn
The installation-defined message-chaining transaction identification code (for example, C/), as defined in the PGCHAIN system initialization parameter.

Yyyy
The identifier of a transaction that produces data to be added to the message chain.

Parameters
The parameters, if any, required by transaction yyyy.

If the new transaction uses terminal control, or BMS in TERMINAL mode, its output is displayed immediately. When the new transaction terminates, anything else you enter is interpreted as a paging command for the first message. For example:

TRAN The transaction invokes an application program that produces multiple pages of output. The first page is displayed.

P/2 The second page of message is displayed.

C/CEMT INQ TIM
CEMT data is displayed.

P/2 The second page of message is redisplayed.

If the new transaction uses BMS in PAGING mode, and RETAIN or RELEASE is specified in the SEND PAGE command in the application program, the output is chained to existing output, and a new level of chaining is created.

If RELEASE is specified, this chaining process can be repeated indefinitely. However, chaining requests must always be made while viewing pages at the highest level of chaining. Otherwise, messages at levels higher than the current level is automatically purged (if OPERPURGE has not been specified in the application program SEND PAGE command), or you are asked to purge them. If RETAIN is specified, no further chaining (in either TERMINAL or PAGING mode) can be performed until the second logical message has been terminated. Further chaining requests are rejected.

If neither RETAIN nor RELEASE is specified, the output from the second transaction is queued as a new message for the terminal, and is not received until all earlier messages have been terminated. It is thus not chained to the current message.

If you enter data that is not a paging command during a message chaining session, all pages on levels higher than the current level are deleted, if no message in the chain has been built using OPERPURGE=YES in the SEND PAGE command.

An example of the use of message chaining under BMS in PAGING mode is a search of a personnel file for people who speak French. Your initial inquiry might turn up three names. Without terminating the page containing the names, you can examine the records of each of the three people named. The chaining transaction could be used to read a record, without erasing the three names.

This would create output at the second level of chaining (the output from the original transaction being first-level output). When you finish viewing a person's record, you enter a paging command to terminate the current level, see "CSPG message termination" on page 559. The message displaying the person's record would be purged, and the original page, containing the three names, redisplayed. Assuming that the message chaining transaction identifier is C/, enter:

```
LANG DEPT=J34,FRENCH
```

Receive:

```
BERESFORD, J.A. 1043
LEVY, A.R.      7121
WALTON, J.      8143
```

To find out more about Beresford, enter:

```
C/PREC 1043
```

This finds the personnel record for employee number 1043. PREC is the identifier of a user-written transaction for examining personnel records. Receive:

```
NAME - BERESFORD, JERRY A.
EMPLOYEE NO - 1043
(and so on)
```

To go back to the list of names, enter:

```
T/C
```

T/ is the user-defined transaction identifier that terminates the message at the current level of chaining (see "CSPG message termination" on page 559).

Receive:

```
BERESFORD, J.A.  1043
LEVY, A.R.       7121
WALTON, J.       8143
```

You can now examine Levy's record.

You can also chain a series of messages together and use paging commands to view the series simultaneously. Using the previous example, you can have the summary page as your base message (first level) and each of the personnel records as subsequent chain levels. Assume you are currently displaying the summary:

Enter:

```
C/PREC 1043
```

Receive:

```
NAME - BERESFORD...
```

Enter:

```
C/PREC 7121
```

Receive:

```
NAME - LEVY...
```

Enter:

```
C/PREC 8143
```

Receive:

```
NAME - WALTON...
```

There are now four levels of logical messages at the terminal. You can now page back and forth between the personnel records. Enter:

```
P/(2)1
```

Receive:

```
NAME - BERESFORD...
```

Enter:

```
P/(L)1
```

Receive:

```
NAME - WALTON...
```

You can chain a new transaction only while you are viewing the last logical level. If you have four levels (as in the above example), view the summary page (level one) and then enter C/PREC 3642; the new transaction becomes level two and all other chained messages below that are erased (assuming that they have been built using a SEND PAGE command with the OPERPURGE option omitted).

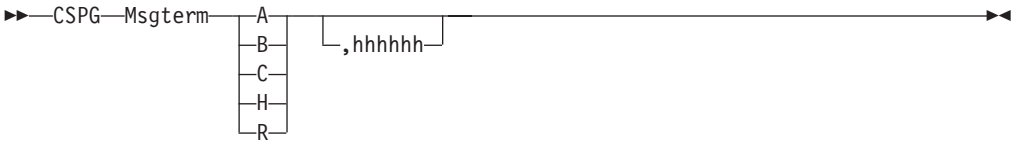
Messages involving different partition layouts (different partition sets, including unpartitioned) may be chained. The current page in each partition is redisplayed on a change of message level.

CSPG message termination

You can erase a logical message using the message terminate command.

If the logical message is security protected, it is terminated only if your identification or class matches that in the message. After an erase, the last message still in the chain is displayed. The format of the message terminate command is as follows:

CSPG MSGTERM



Msgterm
A user-defined transaction identifier (1–7 characters) for the message termination transaction (for example T/) as defined in the PGPURGE system initialization parameter.

A Terminates all logical messages destined for and being displayed on that terminal (including routed messages not yet displayed). The following message is received when termination has been completed:

```
DFHTP4122 Requested purge completed successfully
```

B Terminates the logical message being displayed on that terminal and all logical messages chained to it. This does not include routed messages. Whenever you have finished with all the output from a transaction, you should terminate the base message by entering a command of the form `msgtermB`. If the message

has been built using a SEND PAGE command with the OPERPURGE option omitted, you can terminate the paging session by entering data other than a paging command.

C Terminates the current (level) logical message and any messages chained below it.

H Terminates all logical messages chained to the base logical message being displayed on that terminal. The page of the base message that was most recently displayed is redisplayed at the terminal.

R Terminates all logical messages queued for immediate delivery (routed) to the terminal. The page that was being displayed at the terminal is redisplayed.

Hhhhhh
The 6-digit hexadecimal identifier of a specific, undelivered logical message to be erased. (This identifier would have been determined using a page query command.) If the message specified is security-protected, the operator should have a security code that matches the one in the message.

If a message identifier (hhhhhh) is not specified in the termination request and there is no message currently displayed, the oldest message is automatically retrieved.

Copying pages with CSPG

You can send a copy of the page being displayed, as it appeared when initially transmitted, to another terminal (normally hardcopy).

You do this as follows:

Enter the page-copy transaction identifier and the identifier of the terminal that is to receive the copy, as follows:

```
msgcopytttt
```

where:

Msgcopy
The copy transaction identifier (1–7 characters), for example D/, as defined in the **PGCOPY** system initialization parameter.

Tttt
The identifier of the terminal to which the page is to be copied.

You get the message:

```
DFHTP4124 Page copied from terminal tttt
        (message number msgno)
```

Field and character attributes in the copied page that are not supported by the target terminal are ignored. The copied page is reformatted if it does not fit on the target terminal. Field and character attributes are discarded in the reformatting. If partitions are in use, the page copy command copies the displayed page for the partition in which the copy command is entered.

Using Autopage with CSPG

A hardcopy terminal that is not defined as a paging terminal (one that displays pages when you request them) is an autopaging terminal (one that displays all pages without your intervention).

An autopaging terminal may be temporarily designated as a paging terminal by the user's application program using the NOAUTOPAGE option of the SEND PAGE command. If this is done at the time the first page is to be displayed on the terminal, the terminal is temporarily treated as a paging terminal for the duration of the message.

This condition exists until either of the following occurs:

1. You enter P/A (reset terminal to autopaging status), indicating that all subsequent pages are to be displayed without further intervention.
2. You enter T/C (terminate current message). All pages from that logical message are purged, and the terminal is again put into autopaging status.

P/ is the paging transaction identifier, and T/ is the terminate transaction identifier.

Displaying routed pages with CSPG

You can use the message-switching transaction to direct output to a terminal or terminals other than your own. For more information on message switching, see Chapter 26, "CMSG - message switching," on page 511. When a terminal to which you have directed a message is in TRANSCEIVE status, and is not involved with a transaction, and is not displaying pages, CICS displays the first page of that message.

However, if the terminal is in TRANSACTION status, you can display the first page only of a waiting message by entering CSPG, CSPG P/1, or just P/1. Here, you are using P/ as a paging command. If no page is waiting, you get the following message:

```
DFHTP4104 A paging request has been received but
    there are no pages for display
```

Chapter 31. CWTO - write to console operator

Use CWTO to send messages to the console operator.

Start the transaction by entering CWTO, optionally followed by the text of your message, on the command line of the screen, as follows:
Press the ENTER key. If the transaction runs successfully, your message is sent to

CWTO

the console, and you get the normal completion response:

```
MESSAGE HAS BEEN SENT
```

If you enter `CWTO` with no message text, you are prompted for your message as follows:

```
ENTER MESSAGE
```

You can cancel the CWTO transaction by entering CANCEL as the last 6 characters of your message. The transaction is terminated and you get the following response:

```
TERMINATED BY OPERATOR
```

The following are some general points about the format of your messages:

- If your message exceeds the line length of the console, it is divided between words into more than one line. Each part of a divided message is prefixed by your identifier and by that of the originating terminal.
- New-line (NL) characters within your message are kept, although any leading NL character is deleted.
- You can include blank lines at the top of your transmitted message by pressing ENTER (or using carriage returns) the required number of times.
- You can enter any number of backspace characters for deletion. If you enter only blank characters, or if you delete all of the data by entering too many deletion characters, you get the following message:

  ```
  NO INPUT - REENTER
  ```

© Copyright IBM Corp. 1982, 2011

Chapter 32. DSNC - CICS DB2 transaction DSNC

The DSNC transaction can be used to perform the following:

- Enter DB2 commands from a CICS terminal.
- Cause threads to be terminated when they are released (DSNC DISCONNECT).
- Display information about transactions using the CICS DB2 interface, and display statistics (DSNC DISPLAY).
- Modify the unsolicited message destinations, and modify the number of active threads used by a DB2ENTRY, the pool, or for commands (DSNC MODIFY).
- Shut down the CICS DB2 interface (DSNC STOP).
- Start the CICS DB2 interface (DSNC STRT).

For more information about the CICS DB2 transaction, DSNC, see the *CICS DB2 Guide*.

© Copyright IBM Corp. 1982, 2011

Appendix. List of CICS transactions

The following table lists all the transactions that are supplied with CICS, except for those that are part of CICS sample programs.

The transactions that have operator interfaces are marked by an asterisk (*). The remainder therefore have no operator interface.

Transaction is the transaction identifier. **Program** indicates the program (if there is one) that is specified in the supplied transaction definition. **CSD group** indicates the supplied resource-definition group that contains the transaction definition. For general information on resource definition and the CICS groups, see the *CICS Resource Definition Guide*.

Table 11. CICS transactions supplied by IBM

Transaction	Program	CSD group	Security category	Description
CADP*	DFHDPLU	DFHDP	2	Application debugging profile manager
CAQP	DFHAQP			Asynchronous Queue Purge Program
CATA	DFHZATA	DFHSPI	1	Defines autoinstall automatic terminal
CATD	DFHZATD	DFHSPI	1	Deletes autoinstall terminal
CATP	DFHATP			Asynchronous Transaction Control Program
CATR	DFHZATR	DFHSPI	3	Deletes autoinstall restart terminal
CATS	DFHZATS			
CAUT	DFHSTSP			Automatic System Statistics Program
CBAM*	DFHECBAM	DFHOPER	2	BTS objects browser
CBRA	DFHSTSP			Bridge facility allocation
CBRC	DFHBRCP	DFHDLI		
CCIN	DFHZCN1	DFHCLNT	3	CICS Client
CCMF	DFHCCMF			Monitoring ATI Program
CCMS	EFHCCMS			CMS Control Transaction
CCRL*	DFHSOCRL	DFHOPER	2	CICS certificate revocation list transaction
CCU2	EFHUCMS2			CMS VSAM Conversion utility
CDBC*	DFHDBME	DFHDBCTL	2	DBCTL interface menu transaction
CDBD	DFHDBDI	DFHDBCTL	1	DBCTL disable function
CDBF	DFHD2CM3	DFHDB2	1	CICS DB2 attachment facility shutdown force transaction
CDBI*	DFHDBIQ	DFHDBCTL	2	DBCTL interface inquiry transaction
CDBM*	DFHDBMP	DFHDBCTL	2	DBCTL operator transaction
CDBN	DFHDBCON	DFHDBCTL	Exempt (See note)	DBCTL issue commands
CDBO	DFHDBCT	DFHDBCTL	1	DBCTL control function
CDBQ	DFHD2CM2	DFHDB2	1	CICS DB2 attachment facility shutdown quiesce transaction
CDBT	DFHDBDSC	DFHDBCTL	2	DBCTL interface disconnection transaction
CDFS	DFHDFST	DFHISC	2	Dynamic starts with interval
CDST	DFHMIRS			CICS Dynamic routing of non-term starts
CDTS	DFHZATS	DFHSPI	1	Provides remote single delete transaction
CEBR*	DFHEDFBR	DFHEDF	2	Browse temporary storage
CEBT*	-	-		Master terminal, alternate CICS
CECI*	DFHECIP	DFHINTER	2	Command level interpreter
CECS*	DFHECSP	DFHINTER	2	Command level interpreter
CEDA*	DFHEDAP	DFHSPI	2	Resource definition online - full (RDO)

© Copyright IBM Corp. 1982, 2011

Table 11. CICS transactions supplied by IBM (continued)

Transaction	Program	CSD group	Security category	Description
CEDB*	DFHEDAP	DFHSPI	2	Resource definition online - restricted (RDO)
CEDC*	DFHEDAP	DFHSPI	2	Views resource definition online (RDO)
CEDF*	DFHEDFP	DFHEDF	2	Execution diagnostic facility
CEDX*	DFHEDFP	DFHEDF	2	Execution diagnostic facility for non-terminal tasks
CEGN	DFHCEGN	DFHSIGN	3	Schedules good night transaction
CEHP	DFHCHS	DFHISC	2	CICS OS/2 request handler
CEHS	DFHCHS	DFHISC	2	CICS/VM request handler
CEID	DFHCEID			UUID Timeout Transaction
CEJR	DFHEJITL	DFHSTAND	3	Corbaserver resolution
CEKL*			Exempt (See note)	Master terminal for emergency use
CEMN	DFHCEMNA	DFHOPER	2	CICS monitoring facility transaction
CEMS	DFHEMSP			Spooling Master Terminal Transaction
CEMT*	DFHEMTP	DFHOPER	2	Master terminal
CEOS	DFHEOSP			Spooling Terminal Operator Transaction
CEOT*	DFHEOTP	DFHOPER	2	Terminal status
CEPC	DFHPSCP			Printer Spooling Print Control
CEPE	DFHPSEP			Printer Spooling Report Deletion
CEPQ	DFHPSOP			Print Intra-Partition TD Queue
CEPW	DFHPSOP			Printer spooling writer
CESC*	DFHCESC	DFHSIGN	1	Processes timeout and signoff for idle terminals
CESD	DFHCESD	DFHSDAP	2	Shutdown assist
CESF*	DFHSFP	DFHSIGN	3	Signs off terminal user
CESN*	DFHSNP	DFHSIGN	3	Signs on terminal user
CEST*	DFHESTP	DFHOPER	2	Supervisory terminal
CETR*	DFHCETRA	DFHOPER	2	Inquire and set trace options
CEX2	DFHD2EX2	DFHDB2	1	CICS DB2 protected thread purge mechanism and other CICS DB2 services.
CFCL	DFHFCDL	-	1	CFDT Load
CFOR	DFHFCQT	-	1	RLS offsite recovery
CFQR	DFHFCQT	-	1	RLS quiesce receive
CFQS	DFHFCQT	-	1	RLS quiesce send
CFSL	DFHDTLX	-		
CFTI				Flowmark for MVS
CFTL	DFHDTLX	-	1	Shared DT load
CFTM				Boeblingen Banking Project
CFTS	DFHZATS	DFHSPI	1	Provides remote mass flag transaction
CFUP				Boeblingen Banking Project
CGRP	DFHZCGRP	-	1	Provides VTAM persistent sessions
CHLP	DFHCMAC	DFHCMAC		Alias for CMAC
CIDP*	DFHDPIN	DFHDP	2	Inactivate debugging profiles utility
CIEP	DFHIEP	DFHIPECI	3	ECI for TCP/IP listener
CIND*	DFHINDT	DFHINDT	2	CICS in-doubt testing tool
CINS				
CIOD	DFHIIOPA	DFHIIOP	1	Default IIOP interface, started by CIOR
CIOF	DFHIIOPA	DFHIIOP	1	CICS Generic factory, started by CIOR
CIOR	DFHIIOP	DFHIIOP	1	CICS IIOP interface, started by SO_Domain
CIRB				SQL/DS Transaction ID

Table 11. CICS transactions supplied by IBM (continued)

Transaction	Program	CSD group	Security category	Description
CIRD				SQL/DS Transaction ID
CIRP	DFHIIRP	DFHIIOP	2	Default CICS IIOP request processor transaction
CIRR	DFHIIRRS	DFHIIOP	1	Default CICS IIOP request receiver
CISC	DFHISCOP	DFHISCIP	1	IPIC acquire IPCONN on the client side of a connection
CISD	DFHISCOP	DFHISCIP	1	IPIC release IPCONN on the client side of a connection
CISE	DFHISCIP	DFHISCIP	1	IPIC error and message program
CISR	DFHISRRP	DFHISCIP	1	IPIC request/response receiver
CISS	DFHISCOP	DFHISCIP	1	IPIC acquire IPCONN on the server side of a connection
CIST	DFHISCOP	DFHISCIP	1	IPIC terminate IPCONN
CISX	DFHISREX	DFHISCIP	1	IPCONN recovery and resynchronization transaction for XA clients
CITS	DFHZATS	DFHSPI	1	Provides remote autoinstall transaction
CJGC	DFHSJGC	DFHJAVA	1	CICS JVM garbage collection transaction
CJMJ	DFHSJJM	DFHJAVA	1	Starts the master JVM
CJPI	DFHSJPI	DFHJAVA	1	Starts JVMs following a PERFORM JVMPOOL command
CJTR	DFHOTR	DFHOTS	1	CORBA Object Transaction Services (OTS) resynchronization transaction
CKAM	DFHMQMON	DFHMQ	1	CICS-MQ Adapter alert monitor
CKBM	DFHMQBAS	DFHMQ	2	CICS-MQ Adapter base panel transaction
CKCN	DFHMQQCN	DFHMQ	2	CICS-MQ Adapter start connection transaction
CKBP	CSQCBP00	DFHMQ	2	CICS-MQ Bridge DPL bridge task
CKDL	DFHMQDSL	DFHMQ	2	CICS-MQ Adapter, display status transaction
CKDP	DFHMQDIS	DFHMQ	2	CICS-MQ Adapter, display transaction
CKQC	DFHMQCTL	DFHMQ	2	CICS-MQ Adapter control transaction
CKRS	DFHMQRS	DFHMQ	2	CICS-MQ Adapter modify transaction
CKRT	DFHMQRET	DFHMQ	2	CICS-MQ Adapter screen return transaction
CKSD	DFHMQDSC	DFHMQ	2	CICS-MQ Adapter stop connection transaction
CKSQ	DFHMQSSQ	DFHMQ	2	CICS-MQ Adapter start/stop CKTI transaction
CKTI	DFHMQTSK	DFHMQ	1	CICS-MQ Adapter - task initiator transaction
CLDM*	DFHLDMAP	DFHOPER	2	CICS load module map
CLEP				Language Environment runtime options
CLER				Language Environment runtime options
CLQ2	DFHLUP	DFHISCT	3	Outbound resynchronization for APPC and MRO
CLR1	DFHZLS1	DFHISCT	3	Inbound CNOS for APPC and MRO
CLR2	DFHLUP	DFHISCT	3	Inbound resynchronization for MRO
CLS1	DFHZLS1	DFHISC	3	Provides ISC LU services model
CLS2	DFHLUP	DFHISC	3	Provides ISC LU services model
CLS3	DFHCLS3	DFHISC	3	ISC LU services model
CLS4	DFHCLS4	DFHISC	3	Manages password expiry
CMAC	DFHCMAC		2	Messages utility
CMPX	DFHMXP	DFHISC	3	Ships ISC local queuing
CMSG*	DFHMSP	DFHMSWIT	2	Message switching

Table 11. CICS transactions supplied by IBM (continued)

Transaction	Program	CSD group	Security category	Description
CMTS	DFHZATS	DFHSPI	1	Remote mass delete transaction
COVR	DFHZCOVR	-	1	Provides open VTAM retry transaction
CPIA*	DFHPITE	DFHPIPE	2	Invokes CPIS from the terminal
CPCT	DFHZPCT	DFHPSSGN	3	Catalog signed on terminals for persistent session signon retention
CPIA	DFHPITE	DFHPIPE	2	Invokes CPIS from the terminal
CPIH	DFHPIDSH	DFHPIPE	2	CICS pipeline HTTP inbound router
CPIL	DFHPILSQ	DFHPIPE	2	SOAP WMQ inbound listener
CPIQ	DFHPIDSQ	DFHPIPE	2	SOAP WMQ inbound router
CPIR	DFHPIITL	DFHSTAND	1	Pipeline resolution transaction
CPIS	DFHPIR	DFHPIPE	1	WS-AT transaction that is attached when resynchronization is required
CPLT	DFHSIPLT	-	1	Initializes PLT processing
CPMI	DFHMIRS	DFHISC	2	CICS LU 6.2 synchronization level 1 mirror
CPSS	DFHZSGN	DFHPSSGN	3	Persistent sessions signon
CQPI	DFHCLS5	DFHISC	3	Connection quiesce. Architected transaction (inbound).
CQPO	DFHCLS5	DFHISC	3	Connection quiesce. Architected transaction (outbound).
CQRY	DFHQRY	DFHSTAND	3	Provides ATI query support
CRDR	DFHRD1			Asynchronous Transaction Processing in Put Routine 1
CREA*	DFHADDRM	DFHADST	2	Create REQUESTMODELs for enterprise beans
CREC*	DFHADDRM	DFHADST	2	Create REQUESTMODELs for enterprise beans (read only)
CRMD	DFHZATMD	DFHSPI	1	Provides remote mass delete transaction
CRMF	DFHZATMF	DFHSPI	1	Provides remote mass flag transaction
CRPA	DFHRPAS	DFHRPC	2	ONC/RPC Alias transaction
CRPC	DFHRPC00	DFHRPC	2	ONC/RPC Update transaction
CRPM	DFHRPHS	DFHRPC	2	ONC/RPC Server controller
CRSQ	DFHCRQ	DFHISC	1	Remote schedule purging (ISC)
CRSR	DFHCRS	DFHISC	3	Provides ISC remote scheduler
CRSY	DFHRMSY	DFHRMI	1	Resource manager resynchronization
CRTE*	DFHRTE	DFHISC	2	Transaction routing
CRTP	DFHZRTP	DFHPSSGN	1	Persistent sessions restart timer transaction
CRTX	-	DFHISC	2	Dynamic transaction routing transaction definition
CSAC	DFHACP	DFHSTAND	3	Provides program abnormal condition
CSCY	DFHCPY	DFHVTAMP	3	Provides 3270 screen print
CSFE*	DFHFEP	DFHFE	2	Terminal test, trace, storage
CSFR	DFHFCRD	-	1	RLS cleanup
CSFU	DFHFCU	DFHOPCLS	1	File open utility
CSGM	DFHGMM	DFHVTAM	2	“Good-morning” signon
CSGX	DFHDLG	DFHDLI		
CSHA	-	-	1	Scheduler services (Autoinstalled by CICS)
CSHQ	DFHSHSY	-	1	Scheduler services domain long running task
CSHR	DFHMIRS	DFHISC	2	Scheduler services remote routing
CSIR 160	DFHCRR			Interregion Session Recovery Program
CSJC	DFHJCBSP	DFHCOMP3		Journal control bootstrap

Table 11. CICS transactions supplied by IBM (continued)

Transaction	Program	CSD group	Security category	Description
CSKD	DISABLE			Disable TCP/IP interface adapter
CSKE	ENABLE			Enable TCP/IP interface adapter
CSKL	LISTENER			TCP/IP Listener
CSKP	DFHAKP	DFHCOMP5	1	Writes system log activity keypoint
CSLG	DFHZRLG	DFHRSPLG		
CSMI	DFHMIRS	DFHISC	2	Mirror transaction
CSMT	DFHMTPA			Master Terminal Module A
CSM1	DFHMIRS	DFHISC	2	SYSMSG model
CSM2	DFHMIRS	DFHISC	2	Scheduler model
CSM3	DFHMIRS	DFHISC	2	Queue model
CSM5	DFHMIRS	DFHISC	2	DL/I model
CSNC	DFHCRNP	DFHISC	1	Interregion control program (MRO)
CSNE	DFHZNAC	DFHVTAM	1	Provides VTAM node error recovery
CSOL	DFHSOL	-	1	TCP/IP listener (Autoinstalled by CICS)
CSOT	DFHMTPA			Master Terminal Module A
CSPG*	DFHTPR	DFHBMS	3	Provides BMS terminal paging
CSPK	DFHPRK	DFHVTAMP	3	Provides 3270 screen print support
CSPP	DFHP3270	DFHHARDC	3	Provides 3270 print support
CSPQ	DFHTPQ	DFHBMS	1	Terminal page cleanup (BMS)
CSPS	DFHTPS	DFHBMS	3	Schedules BMS terminal paging
CSQC	DFHLGQC	DFHLGQC	1	CICS quiesce after system log failure
CSRK	DFHRKB	DFHVTAMP	3	Provides 3270 screen print - release keyboard
CSRS	DFHZRSP	DFHRSEND	3	Synchronizes 3614 message
CSSC	DFHCSSC			Sign off inactive terminal
CSSF	DFHRTC	DFHISC	3	Cancels CRTE transaction routing session
CSSN	DFHSNP			Sign on program
CSST	DFHMTPA			Master terminal module A
CSSX	DFHDLS	DFHDLI		
CSSY	DFHAPATT	-	1	Provides entry point attach
CSTA	DFHTAJP			Time of day adjustment program
CSTB				
CSTE	DFHTACP	DFHSTAND	1	Processes terminal abnormal conditions
CSTP	DFHZCSTP	-	1	Provides terminal control transaction
CSTT	DFHSTKC	DFHCOMP1		Supervisory Statistics Program
CSXM			Exempt	The transaction used by CICS services to get and free a transaction environment
CSZI	DFHSZRMP	DFHFEPI	1	Front End Programming Interface (FEPI), only active if FEPI installed.
CTIN	DFHZCT1	DFHCLNT	2	CICS Client
CTSD	DFHTSDQ	-	1	Temporary storage delete recoverable queue
CVMI	DFHMIRS	DFHISC	2	CICS LU6.2 synchronization level 1 mirror
CVST	DFHVAP			Subtask Monitor Program
CWBA	DFHWBA	DFHWEB	2	CICS Web support alias transaction
CWBC	DFHWBC00	DFHWEB		CICS Web support connection manager
CWBG	DFHWBGB	DFHWEB	1	CICS Web support cleanup transaction
CWBM	DFHWBM	DFHWEB		CICS Web support server controller
CWTO*	DFHCWTO	DFHCONS	2	Write to console operator
CWTR	DFHWT1			Asynchronous Transaction Output Phase 1
CWXN	DFHWBXN	DFHWEB	1	CICS Web support attach transaction

Table 11. CICS transactions supplied by IBM (continued)

Transaction	Program	CSD group	Security category	Description
CWXU	DFHWBXN	DFHWEB	1	CICS Web support USER protocol attach transaction
CXCU	DFHCXCU	DFHSTAND	1	Performs XRF tracing catchup
CXRE	DFHZXRE	DFHSTAND	1	Reconnects terminals following XRF takeover
CXRT	DFHCRT	DFHISC	3	Provides Transaction routing relay
DSNC	DFHD2CM1	DFH$DB2	2	DB2 attachment facility transaction

Note:

1. Transactions CGRP, CSSY, CSTP, and CSXM are transaction names that are used by some CICS tasks.
2. Transactions CDBN, CEKL and CSXM are not subject to security checking, and are exempt from security categorization. Any security definitions for these transactions are redundant. **** MISSING FILE ****

Bibliography

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 2:

Memo to Licensees, GI10-2559
CICS Transaction Server for z/OS Program Directory, GI13-0515
CICS Transaction Server for z/OS Release Guide, GC34-6811
CICS Transaction Server for z/OS Installation Guide, GC34-6812
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, GI13-0515
CICS Transaction Server for z/OS Release Guide, GC34-6811
CICS Transaction Server for z/OS Migration from CICS TS Version 3.1, GC34-6858

© Copyright IBM Corp. 1982, 2011

CICS Transaction Server for z/OS Migration from CICS TS Version 1.3, GC34-6855
CICS Transaction Server for z/OS Migration from CICS TS Version 2.2, GC34-6856
CICS Transaction Server for z/OS Installation Guide, GC34-6812

Administration

CICS System Definition Guide, SC34-6813
CICS Customization Guide, SC34-6814
CICS Resource Definition Guide, SC34-6815
CICS Operations and Utilities Guide, SC34-6816
CICS Supplied Transactions, SC34-6817

Programming

CICS Application Programming Guide, SC34-6818
CICS Application Programming Reference, SC34-6819
CICS System Programming Reference, SC34-6820
CICS Front End Programming Interface User's Guide, SC34-6821
CICS C++ OO Class Libraries, SC34-6822
CICS Distributed Transaction Programming Guide, SC34-6823
CICS Business Transaction Services, SC34-6824
Java Applications in CICS, SC34-6825
JCICS Class Reference, SC34-6001

Diagnosis

CICS Problem Determination Guide, SC34-6826
CICS Messages and Codes, GC34-6827
CICS Diagnosis Reference, GC34-6862
CICS Data Areas, GC34-6863-00
CICS Trace Entries, SC34-6828
CICS Supplementary Data Areas, GC34-6864-00

Communication

CICS Intercommunication Guide, SC34-6829
CICS External Interfaces Guide, SC34-6830
CICS Internet Guide, SC34-6831

Special topics

CICS Recovery and Restart Guide, SC34-6832
CICS Performance Guide, SC34-6833
CICS IMS Database Control Guide, SC34-6834
CICS RACF Security Guide, SC34-6835
CICS Shared Data Tables Guide, SC34-6836
CICS DB2 Guide, SC34-6837
CICS Debugging Tools Interfaces Reference, GC34-6865

CICSPlex SM books for CICS Transaction Server for z/OS

General

CICSPlex SM Concepts and Planning, SC34-6839
CICSPlex SM User Interface Guide, SC34-6840
CICSPlex SM Web User Interface Guide, SC34-6841

Administration and Management

CICSPlex SM Administration, SC34-6842
CICSPlex SM Operations Views Reference, SC34-6843
CICSPlex SM Monitor Views Reference, SC34-6844
CICSPlex SM Managing Workloads, SC34-6845
CICSPlex SM Managing Resource Usage, SC34-6846
CICSPlex SM Managing Business Applications, SC34-6847

Programming

CICSPlex SM Application Programming Guide, SC34-6848
CICSPlex SM Application Programming Reference, SC34-6849

Diagnosis

CICSPlex SM Resource Tables Reference, SC34-6850
CICSPlex SM Messages and Codes, GC34-6851
CICSPlex SM Problem Determination, GC34-6852

CICS family books

Communication

CICS Family: Interproduct Communication, SC34-6853
CICS Family: Communicating from CICS on zSeries, SC34-6854

Licensed publications

The following licensed publications are not included in the unlicensed version of the Information Center:

CICS Diagnosis Reference, GC34-6862
CICS Data Areas, GC34-6863-00
CICS Supplementary Data Areas, GC34-6864-00
CICS Debugging Tools Interfaces Reference, GC34-6865

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 2.

Designing and Programming CICS Applications	SR23-9692
CICS Application Migration Aid Guide	SC33-0768
CICS Family: API Structure	SC33-1007
CICS Family: Client/Server Programming	SC33-1435
CICS 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

The reader of this book may also want to refer to the following IBM publications:

- *OS/390 MVS System Commands*, GC28-1781
- *Resource Access Control Facility (RACF): Security Administrator's Guide*, SC28-1340
- *IMS Utilities Reference manual*, SC26-4284
- *IMS Messages and Codes manual*, SC26-4290
- *IMS System Administration Guide*, SC26-4278
- *IMS Operations Guide*, SC26-4287
- *An Introduction to the IBM 3270 Information Display System*, GA27-2739
- *Systems Network Architecture—LU6.2 Reference: Peer Protocols*, SC31-6808
- *MVS/DFP 3.2 System Programming Reference* manual, SC26-4567
- *OS/390 MVS Diagnosis: Tools and Service Aids*, SY28-1085
- *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 1.4.2 Diagnostics Guide*, SC34-6358

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.

© Copyright IBM Corp. 1982, 2011

Index

Special characters

- and CEMT 123
-TERMID*LDC/OPID
  CMSG 525
? and CEMT 123
(value)
  CEMT INQUIRE JMODEL 226
  CEMT INQUIRE JOURNALNAME 227
  CEMT SET PROCESSTYPE 432
* and CEMT 127
*ALL
  CSPG 554
*LDC
  CMSG 525
+ in CEMT syntax 127
+ in scrolling under CEMT 129
+HHMM
  CMSG 533
+M
  CMSG 533
+MM
  CMSG 533
+TERMID*LDC/OPID
  CMSG 525

Numerics

3270 interface
  application debugging profile manager 37

A

abbreviation of keywords
  CEMT transaction 127
abbreviations for keywords 4
ABEND
  CEMT INQUIRE DB2CONN 170, 173
  CEMT INQUIRE DB2ENTRY 177
  CEMT SET DB2CONN 386
  CEMT SET DB2ENTRY 393
access method control block (ACB) 344, 461
ACCESSMETHOD
  CEMT INQUIRE CONNECTION 157
  CEMT INQUIRE DSNAME 193
  CEMT INQUIRE FILE 210
ACCOUNTREC
  CEMT INQUIRE DB2CONN 168
  CEMT INQUIRE DB2ENTRY 175
  CEMT SET DB2CONN 384
  CEMT SET DB2ENTRY 391
ACQ
  CEOT 468
ACQSTATUS
  CEMT INQUIRE MODENAME 238
ACQUIRED
  CEMT INQUIRE CONNECTION 158
  CEMT INQUIRE IPCONN 220
ACQUIRED *(continued)*
  CEMT INQUIRE MODENAME 238
  CEMT INQUIRE NETNAME 246
  CEMT INQUIRE TERMINAL 306
  CEMT SET CONNECTION 376
  CEMT SET IPCONN 414
  CEMT SET MODENAME 424
  CEMT SET NETNAME 428
  CEMT SET TERMINAL 452
acquiring VTAM terminals 428, 452
ACTION
  CEMT INQUIRE DSNAME 193
  CEMT INQUIRE SYDUMPCODE 272
  CEMT INQUIRE TRDUMPCODE 314
  CEMT SET TSQUEUE 458
activating debugging profiles
  with the 3270 interface 40
  with the web interface 23
ACTIVE
  CEMT INQUIRE ENQ 203
  CEMT INQUIRE MODENAME 238
  CEMT INQUIRE SYSTEM 279, 280
  CEMT INQUIRE TCLASS 288
  CEMT INQUIRE UOW 328
  CEMT INQUIRE UOWENQ 336
ACTIVITY
  CEMT INQUIRE TASK 283
  CEMT SET PROCESSTYPE 431
ACTIVITYID
  CEMT INQUIRE TASK 283
ACTJVMTCBS
  CEMT INQUIRE DISPATCHER 182
ACTOPENTCBS
  CEMT INQUIRE DISPATCHER 182
ACTSOCKETS
  CEMT INQUIRE TCPIP 289
ACTSSLTCBS
  CEMT INQUIRE DISPATCHER 182
ACTXPTCBS
  CEMT INQUIRE DISPATCHER 182
ADD
  CEMT INQUIRE SYDUMPCODE 272
  CEMT INQUIRE TRDUMPCODE 314
  CEMT SET SYDUMPCODE 439
  CEMT SET TRDUMPCODE 456
ADDABLE
  CEMT INQUIRE FILE 210
  CEMT SET FILE 407
ADDREXECENQ
  CEMT INQUIRE ENQ 204
  CEMT INQUIRE UOWENQ 336
ADDSTATUS
  CEMT INQUIRE FILE 210
AFFINITY
  CEMT INQUIRE CONNECTION 157
AFTER
  CEDA 99

© Copyright IBM Corp. 1982, 2011

AGE
CEMT INQUIRE JVM 230
CEMT INQUIRE UOW 325
AGING
CEMT INQUIRE DISPATCHER 182
CEMT INQUIRE SYSTEM 275
CEMT SET DISPATCHER 397
CEMT SET SYSTEM 441
AIBRIDGE
CEMT INQUIRE AUTOINSTALL 141
CEMT SET AUTOINSTALL 371
AKP
CEMT INQUIRE SYSTEM 275
CEMT SET SYSTEM 441
ALL
CEBT INQUIRE AUXTRACE 80
CEBT SET AUXTRACE 84
CEDA 99
CEMT INQUIRE AUTINSTMODEL 140
CEMT INQUIRE AUXTRACE 144
CEMT INQUIRE BEAN 146
CEMT INQUIRE BRFACILITY 148
CEMT INQUIRE CONNECTION 155
CEMT INQUIRE CORBASERVER 165
CEMT INQUIRE DB2CONN 167
CEMT INQUIRE DB2ENTRY 174
CEMT INQUIRE DB2TRAN 178
CEMT INQUIRE DJAR 184
CEMT INQUIRE DSNAME 191
CEMT INQUIRE ENQMODEL 206
CEMT INQUIRE FILE 208
CEMT INQUIRE IPCONN 220
CEMT INQUIRE JMODEL 226
CEMT INQUIRE JOURNALNAME 227
CEMT INQUIRE MODENAME 237
CEMT INQUIRE NETNAME 242
CEMT INQUIRE PARTNER 247
CEMT INQUIRE PIPELINE 249
CEMT INQUIRE PROCESSTYPE 252
CEMT INQUIRE PROFILE 253
CEMT INQUIRE PROGRAM 255
CEMT INQUIRE REQUESTMODEL 263
CEMT INQUIRE STREAMNAME 269
CEMT INQUIRE SYDUMPCODE 271
CEMT INQUIRE TASK 281
CEMT INQUIRE TCLASS 287
CEMT INQUIRE TDQUEUE 296
CEMT INQUIRE TERMINAL 301
CEMT INQUIRE TRANSACTION 308
CEMT INQUIRE TRDUMPCODE 313
CEMT INQUIRE TSMODEL 316
CEMT INQUIRE TSPOOL 318
CEMT INQUIRE TSQUEUE 320
CEMT INQUIRE UOW 324
CEMT INQUIRE UOWLINK 337
CEMT INQUIRE URIMAP 341
CEMT INQUIRE WEBSERVICE 348
CEMT INQUIRE WORKREQUEST 351
CEMT PERFORM STATISTICS 367
CEMT SET AUXTRACE 372
CEMT SET BRFACILITY 374
ALL *(continued)*
CEMT SET CONNECTION 376
CEMT SET CORBASERVER 382
CEMT SET DOCTEMPLATE 398
CEMT SET DSNAME 401
CEMT SET FILE 407
CEMT SET IPCONN 414
CEMT SET JOURNALNAME 419
CEMT SET MODENAME 424
CEMT SET NETNAME 428
CEMT SET PIPELINE 430
CEMT SET PROCESSTYPE 432
CEMT SET PROGRAM 433
CEMT SET SYDUMPCODE 439
CEMT SET TASK 445
CEMT SET TCLASS 446
CEMT SET TERMINAL 452
CEMT SET TRANSACTION 455
CEMT SET TRDUMPCODE 457
CEMT SET TSQUEUE 458
CEMT SET UOW 459
CEMT SET UOWLINK 460
CEMT SET WEBSERVICE 464
CEMT SET WORKREQUEST 465
CMSG 521
general statement 127
ALL ROUTE ENTRIES DELETED
CMSG 527
ALL*LDC
CMSG 525
ALLOCAGE
CEMT INQUIRE JVM 230
ANALYZER
CEMT INQUIRE URIMAP 342
ANALYZERSTAT
CEMT INQUIRE URIMAP 342
ANY
CEMT INQUIRE PROGRAM 259
CEMT INQUIRE TRANSACTION 312
APIST
CEMT INQUIRE PROGRAM 257
APPC
CEMT INQUIRE CONNECTION 160
CEMT INQUIRE UOWLINK 339
APPENDCRLF
CEMT INQUIRE DOCTEMPLATE 187
application debugging profile manager 17
3270 interface 37
activating debugging profiles 40
changing debugging profiles 41
copying debugging profiles 41
create Java debugging profile screen 47
create LE debugging profile screen 43
creating debugging profiles 39
deleting debugging profiles 42
inactivating debugging profiles 40
list debugging profiles screen 37
set LE debugging display device screen 54
setting the display device 53
view Java debugging profile screen 51
view LE debugging profile screen 50

application debugging profile manager *(continued)*
3270 interface *(continued)*
viewing debugging profiles 41
web interface 17
activating debugging profiles 23
changing debugging profiles 24
configuring 17
copying debugging profiles 24
Create Java profile page 29
Create LE profile page 25
creating debugging profiles 22
deleting debugging profiles 25
inactivating debugging profiles 23
List profiles page 19
Set LE display device page 35
setting the display device 34
View Java profile page 33
View LE profile page 32
viewing debugging profiles 24
APPLID
CEMT INQUIRE IPCONN 220
APPLNAME
CEMT INQUIRE MONITOR 239
APPLNAMEST
CEMT INQUIRE MONITOR 239
AS
CEDA 99
ASSEMBLER
CEMT INQUIRE PROGRAM 260
ASSERTED
CEMT INQUIRE CORBASERVER 165
ATI
CEMT INQUIRE NETNAME 243
CEMT INQUIRE TERMINAL 303
CEMT SET NETNAME 428
CEMT SET TERMINAL 452
CEOT 468
ATI (automatic transaction initiation) 300
ATISTATUS
CEMT INQUIRE NETNAME 243
CEMT INQUIRE TERMINAL 303
ATTACHSEC 291
AUDITLEVEL
CEMT INQUIRE PROCESSTYPE 252
AUDITLOG
CEMT INQUIRE PROCESSTYPE 253
AUT
CEOT 468
AUTHENTICATE 292
AUTHID
CEMT INQUIRE DB2CONN 168
CEMT INQUIRE DB2ENTRY 176
CEMT SET DB2CONN 384
CEMT SET DB2ENTRY 392
AUTHTYPE
CEMT INQUIRE DB2CONN 168
CEMT INQUIRE DB2ENTRY 176
CEMT SET DB2CONN 384
CEMT SET DB2ENTRY 392
AUTINSTMODEL
CEMT DISCARD 134
AUTINSTMODEL *(continued)*
CEMT transaction 139
AUTOACTIVE
CEMT INQUIRE SYSTEM 278
CEMT SET SYSTEM 441
AUTOINACTIVE
CEMT INQUIRE SYSTEM 278
CEMT SET SYSTEM 441
AUTOINSTALL
CEMT INQUIRE AUTOINSTALL 141
CEMT PERFORM STATISTICS 368
CEMT SET transaction 370
CEMT transaction 140
CSFE ZCQTRACE 550
AUTOMATIC
CEBT SET TAKEOVER 87
automatic transaction initiation (ATI) 300
AUTOPAGEABLE
CEMT INQUIRE NETNAME 245
CEMT INQUIRE TERMINAL 305
CEMT SET NETNAME 428
CEMT SET TERMINAL 452
CEOT 468
autopaging terminal 561
AUTOPUB
CEMT SET CORBASERVER 382
AUTOPUBLISH
CEMT INQUIRE CORBASERVER 165
AUTOSTARTST
CEMT INQUIRE CLASSCACHE 152
AUTOSWITCH
CEBT INQUIRE DUMP 81
CEBT SET DUMP 85
CEMT INQUIRE DUMPDS 200
CEMT SET DUMPDS 404
AUXILIARY
CEBT INQUIRE AUXTRACE 80
CEMT INQUIRE AUXTRACE 144
CEMT INQUIRE TSMODEL 317
CEMT INQUIRE TSQUEUE 322
auxiliary trace 479
CEBT 79
CEBT SET 83
CEMT 142
CEMT SET 372
AUXTRACE
CEBT 79
CEBT SET transaction 83
CEMT SET transaction 372
CEMT transaction 142
AVAILABLE
CEMT INQUIRE CONNECTION 158
CEMT INQUIRE DSNAME 195
CEMT INQUIRE MODENAME 238
CEMT SET DSNAME 401
CEMT SET MODENAME 424

B

BACKLOG 292
CEMT SET TCPIPSERVICE 448

BACKOUT
CEMT INQUIRE CONNECTION 162
CEMT INQUIRE DSNAME 198
CEMT INQUIRE TASK 285
CEMT INQUIRE TRANSACTION 309
CEMT INQUIRE UOW 326, 327
CEMT SET CONNECTION 376
CEMT SET DSNAME 401
CEMT SET IPCONN 414
CEMT SET UOW 459
BACKUPNONBWO
CEMT INQUIRE UOWDSNFAIL 330
BACKUPTYPE
CEMT INQUIRE DSNAME 195
BASE
CEMT INQUIRE DSNAME 197
BASEDSNAME
CEMT INQUIRE DSNAME 196
basic mapping support (BMS)
page retrieval 553
paging commands 553
BDAM
CEMT INQUIRE DSNAME 193
CEMT INQUIRE FILE 210
BEAN
CEMT PERFORM STATISTICS 368
BEAN command
CEMT transaction 144
BEAN terminal definitions
CEMT INQUIRE BEAN 144
BEANNAME
CEMT INQUIRE REQUESTMODEL 264
BEFORE
CEDA 99
BELOW
CEMT INQUIRE PROGRAM 259
CEMT INQUIRE TRANSACTION 312
BINDING
CEMT INQUIRE WEBSERVICE 348
blank fields in a display 129
BMS (basic mapping support) 553
BMS paging
copy command 560
display routed pages 561
logical message 559
status of hardcopy terminal 561
BREXIT
CEMT INQUIRE TRANSACTION 309
BRFACILITY
CEMT INQUIRE TASK 284
CEMT SET transaction 373
BRIDGE
CEMT INQUIRE TASK 284
bridge facilities
CEMT INQUIRE requests 146
Bridge facilities
CEMT SET requests 373
BROWSE
CEMT INQUIRE FILE 210
CEMT SET FILE 407
BROWSESTATUS
CEMT INQUIRE FILE 210
BSAM terminals 515
BTS browser transaction 57

C

C
CEMT INQUIRE PROGRAM 260
CACHE
CEMT INQUIRE UOWDSNFAIL 330
CACHEFREE
CEMT INQUIRE CLASSCACHE 152
CACHESIZE
CEMT INQUIRE CLASSCACHE 153
CEMT PERFORM CLASSCACHE command 356
CADP transaction 37, 567
CANCEL
CEMT INQUIRE CONNECTION 160
CEMT SET CONNECTION 376
CEMT SET IPCONN 414
CEMT SET TERMINAL 452
CMSG 516
CANCELED - TEMP STRG ERR N
CMSG 527
canceling a message 512
canceling a transaction 3, 513, 515, 519, 563
CAQP transaction 567
CATA transaction 567
CATD transaction 567
CATP transaction 567
CATR transaction 567
CATS transaction 567
CAUSE
CEMT INQUIRE UOWDSNFAIL 330
CAUT transaction 567
CAVM (CICS availability manager) 88
CBAM transaction 57, 567
CBRA transaction 567
CBRC transaction 567
CCIN transaction 567
CCMF transaction 567
CCMS transaction 567
CCRL transaction 59, 567
CCSID
CEMT INQUIRE WEBSERVICE 348
CCU2 transaction 567
CDATAKEY
CEMT INQUIRE TRANSACTION 311
CDBC transaction 61, 567
CDBD transaction 567
CDBF transaction 567
CDBI transaction 65, 567
CDBM transaction 67, 567
CDBN transaction 567
CDBO transaction 567
CDBQ transaction 567
CDBT transaction 567
CDFS transaction 567
CDSASIZE
CEMT INQUIRE DSAS 189

CDST transaction 567
CDTS transaction 567
CEBR transaction 75, 567
CEBT transaction 567
  AUXTRACE 79, 83
  CICS availability manager (CAVM) 88
  DUMP 80, 84
  GTFTRACE 81, 85
  INQ|SET 79
  INTTRACE 82, 86
  PERFORM 88
  request formats 79
  SHUTDOWN 88
  SNAP 88
  SURVEILLANCE 83, 87
  TAKEOVER 83, 87, 89
  XRF complex 79
CECI transaction 91, 567
CECS transaction 93, 567
CEDA transaction 95, 567
  security checking 103
CEDB transaction 95, 568
CEDC transaction 95, 568
CEDF
  CEMT INQUIRE PROGRAM 258
  CEMT SET PROGRAM 433
CEDF transaction 109, 568
CEDFSTATUS
  CEMT INQUIRE PROGRAM 258
CEDX transaction 568
CEGN transaction 568
CEHP transaction 568
CEHS transaction 568
CEID transaction 568
CEJR transaction 568
CEKL transaction 568
  request formats 115
  TASK 115, 118
CEMN monitoring facility transaction 121
CEMN transaction 121, 568
CEMS transaction 568
CEMT INQUIRE CONNECTION 162
CEMT PERFORM CLASSCACHE command 354
CEMT PERFORM JVMPOOL command 362
CEMT PERFORM PIPELINE command 363
CEMT transaction 568
  - character 123
  ? character 123
  * symbol 127
  + in CEMT syntax 127
  + in scrolling 129
  abbreviation of keywords 127
  AUTINSTMODEL 139
  AUTOINSTALL 140, 370
  AUXTRACE 142, 372
  BEAN 144
  blank fields in a display 129
  BRFACILITY 146, 373
  CEMT as a prompter 124
  CEMT on the command line 123
  CFDTPOOL 150
CEMT transaction *(continued)*
  CLASSCACHE 151, 374
  CONNECTION 154, 375
  CORBASERVER 163, 381
  DB2CONN 167, 383
  DB2ENTRY 390
  DB2TRAN 178, 394
  DBENTRY 174
  DELETSHIPPED 180, 395
  DISCARD 132, 134
  DISPATCHER 181, 396
  DJAR 183
  DOCTEMPLATE 186, 398
  DSAS 188, 399
  DSNAME 191, 400
  DUMP 360
  DUMPDS 199, 404
  ENDAFFINITY 361
  ENQ 200
  ENQMODEL 204
  ENQQUEUE 405
  EXCI 206
  Family of resources 127
  FILE 208, 406
  generic names 127
  generic resource identifiers 127
  GTFTRACE 215, 411
  HOST 216, 411
  INQUIRE 123
  INQUIRE IPCONN 219
  INQUIRE LIBRARY 232
  INTTRACE 218, 412
  invoking from an application program 132
  IPCONN 413
  IRBATCH 224
  IRC 224, 417
  JMODEL 225
  JOURNALNAME 227, 418
  JVM 229
  JVMPOOL 231, 419
  LIBRARY 420
  LINE 235, 422
  minus sign following CEMT 123
  MODENAME 236, 423
  MONITOR 238, 424
  NETNAME 241, 427
  overtyping a display 128
  PARTNER 247
  PERFORM 354
  PIPELINE 429
  PROCESSTYPE 431
  PROFILE 253
  PROGRAM 254, 432
  program function (PF) keys 126
  question mark preceding CEMT 123
  QUEUE 435
  request formats 123
  REQUESTMODEL 262
  RESET 364
  RRMS 265
  scrolling a display 129

CEMT transaction *(continued)*
SECURITY 364
security checking 132
SET 123
SHUTDOWN 365
SNAP 360
STATISTICS 265, 366, 435
STREAMNAME 269
SYDUMPCODE 270, 437
SYSTEM 274, 440
tab key 129
TASK 280, 444
TCLASS 287, 445
TCPIP 288, 446
TCPIPSERVICE 290, 448
TDQUEUE 295, 449
TERMINAL 300, 451
TRANSACTION 307, 454
TRDUMPCODE 312, 456
TSMODEL 315
TSPOOL 318
TSQNAME 319
TSQUEUE 319, 457
UOW 323, 458
UOWDSNFAIL 328
UOWENQ 333
UOWLINK 337, 459
URIMAP 340, 460
VTAM 344, 461
WEB 346, 463
WEBSERVICE 463
without keywords 126
WORKREQUEST 350, 464
CEOS transaction 568
CEOT transaction 467, 568
CEPC transaction 568
CEPE transaction 568
CEPQ transaction 568
CEPW transaction 568
CERTIFICATE 292
CEMT INQUIRE CORBASERVER 165
CEMT INQUIRE URIMAP 343
CERTIFICATE option
CEMT INQUIRE IPCONN command 221
certificate revocation list transaction (CCRL) 59
CESC transaction 568
CESD transaction 568
CESF transaction 471, 568
CESN 475, 476
CESN transaction 473, 568
CEST transaction 477, 568
CETR transaction 479, 568
CEX2 transaction 568
CEXECKEY
CEMT INQUIRE PROGRAM 260
CEMT PERFORM JVMPOOL command 362
CFCL transaction 568
CFDTPOOL
CEMT INQUIRE CFDTPOOL 151
CEMT SET FILE 407
CEMT transaction 150
CFOR transaction 568
CFQR transaction 568
CFQS transaction 568
CFSL transaction 568
CFTABLE
CEMT INQUIRE FILE 215
CEMT INQUIRE UOWLINK 340
CEMT SET FILE 407
CFTI transaction 568
CFTL transaction 568
CFTM transaction 568
CFTS transaction 568
CFUP transaction 568
CGROUP
CEMT INQUIRE DB2CONN 169
CEMT SET DB2CONN 385
CGRP transaction 568
chaining a message
CSPG transaction 557
partitions 9
changing debugging profiles
with the 3270 interface 41
with the web interface 24
CHARACTERSET
CEMT INQUIRE URIMAP 344
CHLP transaction 568
CICS availability manager (CAVM) 88
CICS DB2 transaction (DSNC) 565
CICS releases, CSD compatibility across 101
CICSAPI
CEMT INQUIRE PROGRAM 257
CICSTABLE
CEMT INQUIRE FILE 215
CEMT SET FILE 407
CICSTELEVEL
CEMT INQUIRE SYSTEM 275
CID
CEMT INQUIRE NETNAME 243
CEMT INQUIRE TERMINAL 303
CIDDOMAIN
CEMT INQUIRE PIPELINE 249
CIDP transaction 497, 568
CIEP transaction 568
CIND transaction 499, 568
running 500
CINS transaction 568
CIOD transaction 568
CIOF transaction 568
CIOR transaction 568
CIPHERS
CEMT INQUIRE URIMAP 343
CIPHERS option
CEMT INQUIRE IPCONN command 221
CIRB transaction 568
CIRD transaction 569
CIRP transaction 569
CIRR transaction 569
CISC transaction 569
CISE transaction 569
CISR transaction 569
CISS transaction 569

CIST transaction 569
CISX transaction 569
CITS transaction 569
CJGC transaction 569
CJMJ transaction 569
CJPI transaction 569
CJTR transaction 569
CKAM transaction 569
CKBM transaction 569
CKBP transaction 569
CKCN transaction 569
CKDL transaction 569
CKDP transaction 569
CKOPEN
  CEMT INQUIRE TASK 286
CKRS transaction 569
CKRT transaction 569
CKSD transaction 569
CKSQ transaction 569
CKTI transaction 569
CLASS
  CEMT INQUIRE PROGRAM 255
  CEMT INQUIRE TERMINAL 301
  CEMT INQUIRE TRANSACTION 308
  CEMT SET PROGRAM 433
  CEMT SET TERMINAL 453
  CEMT SET TRANSACTION 455
  general statement 127
class tasks, CEMT requests 287, 445
CLASSCACHE
  CEMT SET CLASSCACHE 374
  CEMT transaction 151, 354
CLASSCACHEST
  CEMT INQUIRE JVM 230
CLDM transaction 503, 569
CLEAR key
  when invoking transaction 3
  when used with partitions 8
CLEAR PARTITION key 8
CLEAR/CANCL or CONT INPUT/ROUTE/MSG
  CMSG 528
CLEP transaction 569
CLER transaction 505, 569
CLIENT
  CEMT INQUIRE URIMAP 342
CLIENTCERT
  CEMT INQUIRE CORBASERVER 165
CLIENTIPADDR option
  INQUIRE WORKREQUEST command 352
CLOSED
  CEBT INQUIRE DUMP 81
  CEBT SET DUMP 85
  CEMT INQUIRE DUMPDS 200
  CEMT INQUIRE FILE 213
  CEMT INQUIRE IRC 225
  CEMT INQUIRE MODENAME 238
  CEMT INQUIRE RRMS 265
  CEMT INQUIRE TDQUEUE 299
  CEMT INQUIRE VTAM 346
  CEMT SET DUMPDS 405
  CEMT SET FILE 407
CLOSED *(continued)*
  CEMT SET IRC 418
  CEMT SET MODENAME 424
  CEMT SET VTAM 461
CLOSETIMEOUT
  INQUIRE TCPIPSERVICE 293
CLQ2 transaction 569
CLR1 transaction 569
CLR2 transaction 569
CLS1 transaction 569
CLS2 transaction 569
CLS3 transaction 569
CLS4 transaction 569
CMAC transaction 507, 569
CMDPROT
  CEMT INQUIRE SYSTEM 275
CMDPROTECT
  CEMT INQUIRE SYSTEM 275
CMPX transaction 569
CMSG transaction 511, 569
  CANCEL 516
  DATE 516
  ERRTERM 518
  FULLDATE 518
  HEADING 519
  ID 519
  logical unit destination 524
  MSG 519
  OPCLASS 520
  options 516
  program (DFHMSP) 527
  PROTECT 520
  ROUTE 520
  SEND 523
  syntax 515
  TIME 523
CMTS transaction 570
COBOL
  CEMT INQUIRE PROGRAM 260
codes, display 507
COLD
  CEMT INQUIRE UOWLINK 339
COLDACQ
  CEMT INQUIRE NETNAME 246
  CEMT INQUIRE TERMINAL 307
  CEMT SET NETNAME 428
  CEMT SET TERMINAL 453
COMAUTHID
  CEMT INQUIRE DB2CONN 169
  CEMT SET DB2CONN 385
COMAUTHTYPE
  CEMT INQUIRE DB2CONN 169
COMMAND
  CEBT SET TAKEOVER 87
command interpreter transaction (CECI) 91
command syntax checker (CECS) 93
commands
  CEBT INQ|SET 79
  CEBT PERFORM 88
  CEMT DISCARD 134
  CEMT INQUIRE 138

commands *(continued)*
CEMT PERFORM 354
CEMT SET 370
paging 553
COMMIT
CEMT INQUIRE CONNECTION 162
CEMT INQUIRE DSNAME 198
CEMT INQUIRE TASK 285
CEMT INQUIRE TRANSACTION 309
CEMT INQUIRE UOW 326, 327
CEMT SET CONNECTION 377
CEMT SET DSNAME 401
CEMT SET IPCONN 414
CEMT SET UOW 459
COMMITFAIL
CEMT INQUIRE UOWDSNFAIL 330
communicating with CICS 2
compatibility, CSD, across CICS releases 101
COMPLETE
CEMT INQUIRE CONNECTION 159
component trace options 479
COMPRESS
CEMT INQUIRE MONITOR 239
CEMT SET MONITOR 426
COMPRESSST
CEMT INQUIRE MONITOR 239
COMTHREADLIM
CEMT INQUIRE DB2CONN 170
CEMT SET DB2CONN 386
COMTHREADS
CEMT INQUIRE DB2CONN 170
CONCURRENCY
CEMT INQUIRE PROGRAM 258
CONFIGFILE
CEMT INQUIRE PIPELINE 249
CONNECT
CEMT INQUIRE DB2CONN 173
CEMT SET DB2CONN 389
CONNECTED
CEMT INQUIRE DB2CONN 170
CEMT INQUIRE TSPOOL 319
CEMT SET DB2CONN 386
CONNECTERROR
CEMT INQUIRE DB2CONN 170
CEMT SET DB2CONN 386
CONNECTION
CEDA 99
CEMT DISCARD 134
CEMT INQUIRE CONNECTION 158
CEMT INQUIRE MODENAME 238
CEMT INQUIRE UOW 327
CEMT INQUIRE UOWDSNFAIL 330
CEMT INQUIRE UOWLINK 340
CEMT PERFORM STATISTICS 368
CEMT SET CONNECTION 375
CEMT SET MODENAME 424
CONNECTST
CEMT INQUIRE DB2CONN 170
CEMT SET DB2CONN 386
CONNSTATUS
CEMT INQUIRE CFDTPOOL 151
CONNSTATUS *(continued)*
CEMT INQUIRE CONNECTION 158
CEMT INQUIRE IPCONN 220
CEMT INQUIRE TSPOOL 319
CONSOLE
CEMT INQUIRE TERMINAL 304
console as a CICS terminal 10
console operator, write to 563
console support, multiple 10
CONSOLES
CEMT INQUIRE AUTOINSTALL 141
CEMT SET AUTOINSTALL 371
CONTAINER
CEMT INQUIRE WEBSERVICE 348
CONTENTION
CEMT SET FILE 407
CONTINUE INPUT
CMSG 528
CONTINUE INPUT OR MSG
CMSG 528
CONTINUE MSG
CMSG 528
CONTINUE ROUTE OPERAND
CMSG 528
CONVERSE
CEMT INQUIRE MONITOR 240
CEMT SET MONITOR 426
CONVERSEST
CEMT INQUIRE MONITOR 240
CONVERTER
CEMT INQUIRE URIMAP 343
COORDINATOR
CEMT INQUIRE UOWLINK 339
COPID
CEMT INQUIRE DB2CONN 170
CEMT SET DB2CONN 386
option of DSNCRCT macro 386
COPY command 560
copying debugging profiles
with the 3270 interface 41
with the web interface 24
copying pages
CSPG 560
partitions 9
COPYSTATUS
CEMT INQUIRE PROGRAM 258
CORBASERVER
CEDA 99
CEMT DISCARD 135
CEMT INQUIRE BEAN 146
CEMT INQUIRE DJAR 185
CEMT INQUIRE REQUESTMODEL 264
CEMT PERFORM CORBASERVER 358
CEMT PERFORM STATISTICS 368
CEMT SET CORBASERVER 381
CEMT transaction 163, 357
CORBASERVER option
INQUIRE WORKREQUEST command 352
COVR transaction 570
CPCT transaction 570
CPIA transaction 570

CPIH transaction 570
CPIL transaction 570
CPIQ transaction 570
CPIR transaction 570
CPIS transaction 570
CPLT transaction 570
CPMI transaction 570
CPSS transaction 570
CQP
  CEMT INQUIRE CONNECTION 158
CQPI transaction 570
CQPO transaction 570
CQRY transaction 570
CRDR transaction 570
CRE
  CEOT 468
CREA transaction 535, 570
CREATE
  CEMT INQUIRE NETNAME 244
  CEMT INQUIRE TERMINAL 304
  CEMT SET NETNAME 428
  CEMT SET TERMINAL 453
create Java debugging profile screen
  application debugging profile manager
    3270 interface 47
Create Java profile page
  application debugging profile manager
    web interface 29
create LE debugging profile screen
  application debugging profile manager
    3270 interface 43
Create LE profile page
  application debugging profile manager
    web interface 25
CREATESESS
  CEMT INQUIRE NETNAME 244
  CEMT INQUIRE TERMINAL 304
creating debugging profiles
  with the 3270 interface 39
  with the web interface 22
CREC transaction 535, 570
CRITICAL value 293
CRITICALST option
  INQUIRE LIBRARY command 233
  SET LIBRARY command 421
CRMD transaction 570
CRMF transaction 570
CRPA transaction 570
CRPC transaction 570
CRPM transaction 570
CRSQ transaction 570
CRSR transaction 570
CRSY transaction 570
CRTE transaction 547, 570
CRTX transaction 570
CSAC transaction 570
CSCY transaction 570
CSD compatibility across CICS releases 101
CSFE transaction 549, 570
CSFR transaction 570
CSFU transaction 570
CSGM transaction 570
CSGX transaction 570
CSHA transaction 570
CSHQ transaction 570
CSHR transaction 570
CSIGN
  CEMT INQUIRE DB2CONN 169
  CEMT SET DB2CONN 386
CSIR 160 transaction 570
CSJC transaction 570
CSKD transaction 571
CSKE transaction 571
CSKL transaction 571
CSKP transaction 571
CSLG transaction 571
CSM1 transaction 571
CSM2 transaction 571
CSM3 transaction 571
CSM5 transaction 571
CSMI transaction 571
CSMT transaction 571
CSNC transaction 571
CSNE transaction 571
CSOL transaction 571
CSOT transaction 571
CSPG transaction 553, 571
CSPK transaction 571
CSPP transaction 571
CSPQ transaction 571
CSPS transaction 571
CSQC transaction 571
CSRK transaction 571
CSRS transaction 571
CSSC transaction 571
CSSF transaction 571
CSSN transaction 571
CSST transaction 571
CSSX transaction 571
CSSY transaction 571
CSTA transaction 571
CSTB transaction 571
CSTE transaction 571
CSTP transaction 571
CSTT transaction 571
CSXM transaction 571
CSZI transaction 571
CTERM
  CEMT INQUIRE DB2CONN 169
  CEMT SET DB2CONN 386
  DSNCRCT macro 386
CTIN transaction 571
CTLGALL
  CEMT INQUIRE SYSTEM 277
  CEMT SET SYSTEM 441
CTLGMODIFY
  CEMT INQUIRE SYSTEM 278
  CEMT SET SYSTEM 442
CTLGNONE
  CEMT INQUIRE SYSTEM 278
  CEMT SET SYSTEM 442
CTSD transaction 571

CTX
CEMT INQUIRE DB2CONN 170
CEMT SET DB2CONN 386
CURAUXDS
CEMT INQUIRE AUXTRACE 144
CURRENT
CEMT INQUIRE SYDUMPCODE 273
CEMT INQUIRE TRDUMPCODE 314
CURRENT(A|B)
CEBT INQUIRE AUXTRACE 80
CURRENTDDS
CEBT INQUIRE DUMP 81
CEMT INQUIRE DUMPDS 200
CURREQS
CEMT INQUIRE AUTOINSTALL 141
CUSERID
CEMT INQUIRE DB2CONN 170
CEMT SET DB2CONN 386
CVMI transaction 571
CVST transaction 571
CWBA transaction 571
CWBC transaction 571
CWBG transaction 571
CWBM transaction 571
CWTO transaction 563, 571
CWTR transaction 571
CWXN transaction 571
CWXU transaction 572
CXCU transaction 572
CXRE transaction 572
CXRT transaction 572

D

D
CEMT INQUIRE TASK 286
DAE
CEMT INQUIRE SYDUMPCODE 273
CEMT SET SYDUMPCODE 439
DAEOPTION
CEMT INQUIRE SYDUMPCODE 273
data sets
CEMT requests 208
CEMT SET requests 406
data-value
CEMT INQUIRE LIBRARY 233
database control (DBCTL)
connection to 61
disconnection from 61
database control inquiry transaction (CDBI) 65
database control interface transaction (CDBM) 67
database control menu transaction (CDBC) 61
DATABUFFERS
CEMT INQUIRE TDQUEUE 297
DATALOC
CEMT INQUIRE PROGRAM 259
DATASET
CEMT INQUIRE ENQ 204
CEMT INQUIRE UOW 327
CEMT INQUIRE UOWDSNFAIL 330
CEMT INQUIRE UOWENQ 336
DATASETFULL
CEMT INQUIRE UOWDSNFAIL 331
DATE
CMSG 516
DATE ALREADY PASSED
CMSG 529
DATE INVALID AT XXXXXXXX
CMSG 529
DATE NOT XX/XX AT XXXXX
CMSG 529
date resetting 364
DATE TOO FAR IN FUTURE
CMSG 529
DATESTAMP
CEMT INQUIRE DJAR 185
DATESTARTED
CEMT INQUIRE CLASSCACHE 153
DATFORM, system initialization parameter 8
DB2
CEMT PERFORM STATISTICS 368
DB2 definitions
connection
CEMT INQUIRE DB2CONN 167
CEMT SET DB2CONN command 383
entry
CEMT INQUIRE DB2ENTRY 174
CEMT SET DB2ENTRY command 390
transaction
CEMT INQUIRE DB2TRAN 178
CEMT SET DB2TRAN command 394
DB2CONN
CEDA 99
CEMT DISCARD 135
CEMT INQUIRE DB2CONN 167
CEMT INQUIRE SYSTEM 275
CEMT SET transaction 383
CEMT transaction 167
DB2ENTRY
CEDA 99
CEMT DISCARD 135
CEMT INQUIRE DB2ENTRY 174
CEMT INQUIRE DB2TRAN 179
CEMT SET DB2TRAN 395
CEMT SET transaction 390
CEMT transaction 174
DB2GROUPID
CEMT INQUIRE DB2CONN 170
CEMT SET DB2CONN 387
DB2ID
CEMT INQUIRE DB2CONN 170
CEMT SET DB2CONN 387
DB2PLAN
CEMT INQUIRE TASK 284
DB2RELEASE
CEMT INQUIRE DB2CONN 171
DB2TRAN
CEDA 99
CEMT DISCARD 135
CEMT INQUIRE DB2TRAN 178
CEMT SET transaction 394
CEMT transaction 178

DBCTL (database control) 61
DBCTL system identifier (DBCTLID) 63
DBCTLID 63
DDNAME
  CEMT INQUIRE DOCTEMPLATE 187
  CEMT INQUIRE TDQUEUE 298
DEADLOCK
  CEMT INQUIRE UOWDSNFAIL 331
DEBUG
  CEMT SET SYSTEM 442
debugging profile manager
  3270 interface 37
    activating debugging profiles 40
    changing debugging profiles 41
    copying debugging profiles 41
    create Java debugging profile screen 47
    create LE debugging profile screen 43
    creating debugging profiles 39
    deleting debugging profiles 42
    inactivating debugging profiles 40
    list debugging profiles screen 37
    set LE debugging display device screen 54
    setting the display device 53
    view Java debugging profile screen 51
    view LE debugging profile screen 50
    viewing debugging profiles 41
  web interface 17
    activating debugging profiles 23
    changing debugging profiles 24
    configuring 17
    copying debugging profiles 24
    Create Java profile page 29
    Create LE profile page 25
    creating debugging profiles 22
    deleting debugging profiles 25
    inactivating debugging profiles 23
    List profiles page 19
    Set LE display device page 35
    setting the display device 34
    View Java profile page 33
    View LE profile page 32
    viewing debugging profiles 24
debugging profile manager, application 17
debugging profiles
  activating
    with the 3270 interface 40
    with the web interface 23
  changing
    with the 3270 interface 41
    with the web interface 24
  copying
    with the 3270 interface 41
    with the web interface 24
  creating
    with the 3270 interface 39
    with the web interface 22
  deleting
    with the 3270 interface 42
    with the web interface 25
  inactivating
    with the 3270 interface 40
debugging profiles *(continued)*
  inactivating *(continued)*
    with the web interface 23
  viewing
    with the 3270 interface 41
    with the web interface 24
DEBUGTOOL
  CEMT INQUIRE SYSTEM 275
DEL
  CEMT INQUIRE DELETSHIPPED 180
DELETE
  CEMT INQUIRE FILE 211
  CEMT SET FILE 408
  CEMT SET TSQUEUE 458
  CEMT SET UOWLINK 460
DELETESTATUS
  CEMT INQUIRE FILE 211
deleting debugging profiles
  with the 3270 interface 42
  with the web interface 25
DELETSHIPPED
  CEMT SET transaction 395
  CEMT transaction 359
DELETSHIPPED command
  CEMT transaction 180
DELEXITERROR
  CEMT INQUIRE UOWDSNFAIL 331
DEREGERROR
  INQUIRE TCPIPSERVICE 293
DEREGISTERED
  CEMT INQUIRE VTAM 345
  CEMT SET VTAM 461
  INQUIRE TCPIPSERVICE 293
DEST
  CEMT INQUIRE TASK 284
destinations, multiple 511
DFHCICSP 4
DFLTUSER
  CEMT INQUIRE SYSTEM 276
DISABLED
  CEMT INQUIRE CORBASERVER 165
  CEMT INQUIRE DB2ENTRY 177
  CEMT INQUIRE ENQMODEL 206
  CEMT INQUIRE FILE 212
  CEMT INQUIRE HOST 218
  CEMT INQUIRE JOURNALNAME 229
  CEMT INQUIRE PROGRAM 262
  CEMT INQUIRE TDQUEUE 298
  CEMT INQUIRE TRANSACTION 311
  CEMT INQUIRE URIMAP 342
  CEMT PERFORM CLASSCACHE command 356
  CEMT SET CLASSCACHE command 374
  CEMT SET CORBASERVER 382
  CEMT SET DB2ENTRY 393
  CEMT SET ENQMODEL 406
  CEMT SET FILE 408
  CEMT SET HOST 412
  CEMT SET JOURNALNAME 419
  CEMT SET JVMPOOL 420
  CEMT SET PROCESSTYPE 432
  CEMT SET PROGRAM 433

DISABLED *(continued)*
CEMT SET TRANSACTION 455
CEMT SET URIMAP 460
DISABLEDACT
CEMT INQUIRE DB2ENTRY 177
CEMT SET DB2ENTRY 393
DISABLING
CEMT INQUIRE CORBASERVER 165
CEMT INQUIRE DB2ENTRY 177
DISCARDING
CEMT INQUIRE CORBASERVER 166
DISPATCHABLE
CEKL INQUIRE TASK 116
CEMT INQUIRE TASK 285
DISPATCHER
CEMT INQUIRE transaction 181
CEMT PERFORM STATISTICS 368
CEMT SET transaction 396
display device
setting
with the 3270 interface 53
with the web interface 34
display routed pages 561
display transaction 507
DISPOSITION
CEMT INQUIRE FILE 211
CEMT INQUIRE TDQUEUE 298
DJAR
CEDA 99
CEMT DISCARD 135
CEMT INQUIRE BEAN 146
CEMT PERFORM DJAR 360
CEMT transaction 183, 359
CREA 536
DJARDIR
CEMT INQUIRE CORBASERVER 165
DNSGROUP
INQUIRE TCPIPSERVICE 293
DNSSTATUS
CEMT SET TCPIPSERVICE 448
DNSSTATUS value
INQUIRE TCPIPSERVICE 293
DOCTEMPLATE
CEDA 99
CEMT DISCARD 136
CEMT INQUIRE DOCTEMPLATE 187
CEMT PERFORM STATISTICS 368
CEMT SET transaction 398
CEMT transaction 186
DPLSUBSET
CEMT INQUIRE PROGRAM 260
CEMT SET PROGRAM 433
DROLLBACK
CEMT INQUIRE DB2CONN 171
CEMT INQUIRE DB2ENTRY 177
DS
CEMT INQUIRE TASK 286
DSALIMIT
CEMT INQUIRE DSAS 189
CEMT INQUIRE SYSTEM 276
CEMT SET DSAS 399
DSALIMIT *(continued)*
CEMT SET SYSTEM 442
DSAS
CEMT INQUIRE transaction 188
CEMT SET transaction 399
DSNAME
CEMT INQUIRE DOCTEMPLATE 187
CEMT INQUIRE DSNAME 196
CEMT INQUIRE FILE 211
CEMT INQUIRE TDQUEUE 298
CEMT INQUIRE UOWDSNFAIL 330
CEMT SET FILE 408
CEMT SET transaction 400
CEMT transaction 191
DSNAME option
INQUIRE LIBRARY command 234
DSNC transaction 565, 572
DSNC transactions
DISCONNECT 565
DISPLAY 565
MODIFY 565
STOP 565
STRT 565
DSRTPROGRAM
CEMT SET SYSTEM 442
DTRPROGRAM
CEMT INQUIRE SYSTEM 276
CEMT SET SYSTEM 442
DUMMY
CEMT INQUIRE JMODEL 227
CEMT INQUIRE JOURNALNAME 229
DUMP
CEBT 80
CEBT INQUIRE DUMP 81
CEBT PERFORM SHUTDOWN 88
CEBT PERFORM TAKEOVER 89
CEBT SET transaction 84
CEMT PERFORM DUMP 361
CEMT PERFORM SHUTDOWN 365
CEMT transaction 360
DUMPDS
CEMT INQUIRE DUMPDS 200
CEMT SET transaction 404
CEMT transaction 199
DUMPING
CEMT INQUIRE SYSTEM 276
DUMPSCOPE
CEMT INQUIRE SYDUMPCODE 273
CEMT INQUIRE TRDUMPCODE 314
DURATION
CEMT INQUIRE ENQ 202
CEMT INQUIRE UOWENQ 335
DYNAMIC
CEMT INQUIRE DSNAME 195
CEMT INQUIRE PROGRAM 259
DYNAMSTATUS
CEMT INQUIRE PROGRAM 259

E

ECDSASIZE
  CEMT INQUIRE DSAS 189
EDF (execution diagnostic facility) 109
EDSALIMIT
  CEMT INQUIRE DSAS 189
  CEMT INQUIRE SYSTEM 276
  CEMT SET DSAS 399
  CEMT SET SYSTEM 442
EMPTYREQ
  CEMT INQUIRE FILE 211
  CEMT SET FILE 408
EMPTYSTATUS
  CEMT INQUIRE FILE 211
ENABLED
  CEMT INQUIRE CORBASERVER 166
  CEMT INQUIRE DB2ENTRY 177
  CEMT INQUIRE ENQMODEL 206
  CEMT INQUIRE FILE 212
  CEMT INQUIRE HOST 218
  CEMT INQUIRE JOURNALNAME 229
  CEMT INQUIRE PROGRAM 262
  CEMT INQUIRE TDQUEUE 298
  CEMT INQUIRE TRANSACTION 311
  CEMT INQUIRE URIMAP 342
  CEMT PERFORM CLASSCACHE command 356
  CEMT SET CLASSCACHE command 375
  CEMT SET CORBASERVER 382
  CEMT SET DB2ENTRY 393
  CEMT SET ENQMODEL 406
  CEMT SET FILE 408
  CEMT SET HOST 412
  CEMT SET JOURNALNAME 419
  CEMT SET JVMPOOL 420
  CEMT SET PROCESSTYPE 432
  CEMT SET PROGRAM 433
  CEMT SET TRANSACTION 455
  CEMT SET URIMAP 460
ENABLESTATUS
  CEMT INQUIRE AUTOINSTALL 142
  CEMT INQUIRE CORBASERVER 165
  CEMT INQUIRE DB2ENTRY 177
  CEMT INQUIRE FILE 212
  CEMT INQUIRE HOST 218
  CEMT INQUIRE PIPELINE 249
  CEMT INQUIRE PROCESSTYPE 253
  CEMT INQUIRE TDQUEUE 298
  CEMT INQUIRE URIMAP 342
  CEMT SET DB2ENTRY 393
  CEMT SET PIPELINE 430
ENABLESTATUS option
  INQUIRE LIBRARY command 234
  SET LIBRARY command 421
ENABLING
  CEMT INQUIRE CORBASERVER 166
END
  CSFE 549
ENDAFFINITY
  CEMT INQUIRE CONNECTION 158
  CEMT SET CONNECTION 377
ENDOFDAY
  CEMT INQUIRE STATISTICS 266
  CEMT SET STATISTICS 436
ENDPOINT
  CEMT INQUIRE WEBSERVICE 348
ENQ
  CEMT transaction 200
ENQFAILS
  CEMT INQUIRE ENQ 202
  CEMT INQUIRE UOWENQ 335
ENQMODEL
  CEDA 99
  CEMT DISCARD 136
  CEMT INQUIRE ENQMODEL 206
  CEMT transaction 204
ENQNAME
  CEMT INQUIRE ENQMODEL 206
ENQQUEUE
  CEMT SET transaction 405
ENQSCOPE
  CEMT INQUIRE ENQ 202
  CEMT INQUIRE ENQMODEL 206
  CEMT INQUIRE UOWENQ 335
ENQUEUE
  CEMT PERFORM STATISTICS 368
EQUAL
  CEMT INQUIRE DB2CONN 171, 178
  CEMT SET DB2CONN 388
  CEMT SET DB2ENTRY 393
ERDSASIZE
  CEMT INQUIRE DSAS 190
error message in partition 10
errors in message switching 526
ERRTERM
  CMSG 518
ERRTERM INVALID AT XXXXX
  CMSG 529
ERRTERM INVLD/UNSUPP XXXX
  CMSG 529
ESDSASIZE
  CEMT INQUIRE DSAS 190
EUDSASIZE
  CEMT INQUIRE DSAS 190
EXCEPT
  CEMT INQUIRE MONITOR 240
  CEMT SET MONITOR 426
EXCEPTCLASS
  CEMT INQUIRE MONITOR 240
EXCI
  CEMT INQUIRE CONNECTION 160
  CEMT INQUIRE EXCI 207
  CEMT INQUIRE transaction 206
EXCLSTATUS
  CEMT INQUIRE FILE 212
EXCLUSIVE
  CEMT INQUIRE FILE 212
  CEMT SET FILE 408
EXECENQ
  CEMT INQUIRE ENQ 204
  CEMT INQUIRE UOWENQ 336

EXECKEY
CEMT INQUIRE JVM 231
CEMT INQUIRE PROGRAM 260
execution diagnostic facility (EDF) 109
EXECUTIONSET
CEMT INQUIRE PROGRAM 260
EXTENDED
CEMT INQUIRE FILE 214
EXTRA
CEMT INQUIRE TDQUEUE 300

F

FACILITY
CEMT INQUIRE TASK 284
FACILITYLIKE
CEMT INQUIRE TRANSACTION 309
FAILED
CEMT INQUIRE CONNECTION 159
CEMT INQUIRE JOURNALNAME 229
CEMT INQUIRE STREAMNAME 270
FAILEDBKOUT
CEMT INQUIRE UOWDSNFAIL 331
Family of resources 127
FCANCEL
CEMT INQUIRE CONNECTION 161
CEMT SET CONNECTION 377
FENODE option
CEMT DISCARD 136
FEPI
CEMT PERFORM STATISTICS 368
FEPOOL option
CEMT DISCARD 136
FEPROPSET option
CEMT DISCARD 136
FETARGET option
CEMT DISCARD 136
FILE
CEDA 99
CEMT DISCARD 136
CEMT INQUIRE ENQ 204
CEMT INQUIRE FILE 212
CEMT INQUIRE PROCESSTYPE 253
CEMT INQUIRE UOWENQ 336
CEMT PERFORM STATISTICS 368
CEMT SET transaction 406
CEMT transaction 208
FILECOUNT
CEMT INQUIRE DSNAME 196
FILELIMIT
CEMT INQUIRE MONITOR 240
fixed-format terminals 471
FLUSH
CEMT INQUIRE JOURNALNAME 228
CEMT SET JOURNALNAME 419
FORCE
CEMT INQUIRE SYSTEM 277
CEMT INQUIRE UOW 327
CEMT SET DSNAME 401
CEMT SET SYSTEM 443
CEMT SET UOW 459
FORCECANCEL
CEMT SET IPCONN 415
FORCECLOSE
CEMT INQUIRE FILE 214
CEMT INQUIRE VTAM 346
CEMT SET FILE 408
CEMT SET VTAM 462
FORCEDISABLED
CEMT SET DB2ENTRY 393
FORCENOTCON
CEMT SET DB2CONN 387
FORCEPURGE
CEKL SET TASK 119
CEMT INQUIRE CONNECTION 160
CEMT INQUIRE NETNAME 245
CEMT INQUIRE TASK 285
CEMT INQUIRE TERMINAL 305
CEMT PERFORM CLASSCACHE command 355
CEMT PERFORM JVMPOOL command 362
CEMT SET CONNECTION 377
CEMT SET IPCONN 415
CEMT SET JVMPOOL 420
CEMT SET LINE 422
CEMT SET NETNAME 428
CEMT SET TASK 445
CEMT SET TERMINAL 453
CEMT SET WORKREQUEST 465
INQUIRE WORKREQUEST command 353
FORCEQR
CEMT INQUIRE SYSTEM 277
CEMT SET SYSTEM 443
FORCEUOW
CEMT INQUIRE CONNECTION 162
CEMT SET CONNECTION 378
CEMT SET IPCONN 415
FREEING
CEMT INQUIRE CONNECTION 158
CEMT INQUIRE IPCONN 220
FREQUENCY
CEMT INQUIRE MONITOR 240
CEMT SET MONITOR 426
FRLOG
CEMT INQUIRE DSNAME 196
FTYPE
CEMT INQUIRE TASK 284
FULL
CEMT SET PROCESSTYPE 431
FULLAPI
CEMT INQUIRE PROGRAM 260
CEMT SET PROGRAM 433
FULLDATE
CMSG 518
FWDRECOVABLE
CEMT INQUIRE DSNAME 197
FWDRECOVLSN
CEMT INQUIRE DSNAME 196

G

GARBAGEINT
CEMT INQUIRE WEB 347

GARBAGEINT *(continued)*
CEMT SET WEB 463
generic names 127
generic resource identifiers 127
GETPAGE
CSPG 554
GMT
CEMT INQUIRE MONITOR 241
GRNAME
CEMT INQUIRE CONNECTION 159
CEMT INQUIRE VTAM 345
GROUP
CEDA 99
CEMT INQUIRE DB2ENTRY 176
CEMT SET DB2CONN 384
CEMT SET DB2ENTRY 392
group identifier 475
GROUPID 475
GRPCRITICAL value
INQUIRE TCPIPSERVICE 293
GRSTATUS
CEMT INQUIRE VTAM 345
GTF trace 479
GTFTRACE
CEBT INQUIRE GTFTRACE 82
CEBT INQUIRE transaction 81
CEBT SET transaction 85
CEMT INQUIRE GTFTRACE 216
CEMT INQUIRE transaction 215
CEMT SET transaction 411

H

hardcopy terminal 561
HBACKOUT
CEMT INQUIRE UOW 326
HCOMMIT
CEMT INQUIRE UOW 326
HDISABLED
CEMT INQUIRE URIMAP 342
HEADING
CMSG 519
HEADNG NOT YES/NO AT XXXX
CMSG 529
hexadecimal values 131
HFSFILE
CEMT INQUIRE DOCTEMPLATE 188
CEMT INQUIRE URIMAP 344
HHHHHH
CSPG 554, 560
HHMM
CMSG 523, 533
HHMMSS
CEMT INQUIRE VTAM 346
CEMT SET VTAM 462
HIGH
CEMT INQUIRE DB2CONN 171, 178
CEMT SET DB2CONN 388
CEMT SET DB2ENTRY 393
HOST
CEMT INQUIRE CORBASERVER 166
HOST *(continued)*
CEMT INQUIRE IPCONN 221
CEMT INQUIRE transaction 216
CEMT INQUIRE UOWLINK 338
CEMT INQUIRE URIMAP 343
CEMT SET transaction 411
HOSTCODEPAGE
CEMT INQUIRE URIMAP 344
HTIME
CEMT INQUIRE TASK 284
HTTP
CEMT INQUIRE URIMAP 343
HTTPS
CEMT INQUIRE URIMAP 343
HTYPE
CEMT INQUIRE TASK 284
HVALUE
CEMT INQUIRE TASK 284

I

ID
CMSG 519
ID OPERAND GT 62 CHARS
CMSG 530
ID OPERAND MISSING
CMSG 530
ID PARAM MISSING OR NO)
CMSG 530
IDENTIFIER
CEMT INQUIRE TASK 285
identifiers
resource, lists of 128
identifying terminals 6
IDLE
CEMT INQUIRE DELETSHIPPED 181
CEMT SET DELETSHIPPED 395
IIOP
CEMT INQUIRE UOWLINK 340
CEMT SET WORKREQUEST 465
IMMCLOSE
CEMT INQUIRE IRC 225
CEMT INQUIRE VTAM 346
CEMT SET IRC 418
CEMT SET VTAM 462
IMMEDIATE
CEBT PERFORM SHUTDOWN 88
CEMT PERFORM SHUTDOWN 365
IMMQUIESCED
CEMT INQUIRE DSNAME 197
CEMT SET DSNAME 401
inactivating debugging profiles
with the 3270 interface 40
with the web interface 23
INACTIVE
CEMT INQUIRE SYSTEM 280
INDEXRECFULL
CEMT INQUIRE UOWDSNFAIL 331
INDIRECT
CEMT INQUIRE CONNECTION 157
CEMT INQUIRE TDQUEUE 300

INDOUBT
CEMT INQUIRE TASK 284
CEMT INQUIRE TDQUEUE 298
CEMT INQUIRE TRANSACTION 309
CEMT INQUIRE UOW 326
CEMT INQUIRE UOWDSNFAIL 331
INDOUBTMINS
CEMT INQUIRE TASK 284
CEMT INQUIRE TRANSACTION 309
INDOUBTWAIT
CEMT INQUIRE TASK 285
CEMT INQUIRE TDQUEUE 298
CEMT INQUIRE TRANSACTION 310
INFLIGHT
CEMT INQUIRE UOW 326
Information Display System 515
INPUT
CEMT INQUIRE TDQUEUE 299
CMSG 528
inquire about a resource 123
INQUIRE keyword, CEMT 124
INQUIRE TCPIPSERVICE 291, 292, 293
INS
CEOT 468
INSERVICE
CEMT INQUIRE CONNECTION 162
CEMT INQUIRE IPCONN 223
CEMT INQUIRE NETNAME 246
CEMT INQUIRE TERMINAL 306
CEMT SET CONNECTION 378
CEMT SET IPCONN 415
CEMT SET LINE 423
CEMT SET NETNAME 428
CEMT SET TERMINAL 453
INTERFACE
CEMT INQUIRE REQUESTMODEL 264
interface to master terminal, programmable 132
INTERNAL
CEMT INQUIRE TASK 286
interregion communication (IRC) 224, 417
INTERVAL
CEMT INQUIRE DELETSHIPPED 181
CEMT INQUIRE STATISTICS 267
CEMT SET DELETSHIPPED 396
CEMT SET STATISTICS 436
INTFACETYPE
CEMT INQUIRE REQUESTMODEL 264
INTRA
CEMT INQUIRE TDQUEUE 300
INTTRACE
CEBT INQUIRE INTTRACE 82
CEBT INQUIRE transaction 82
CEBT SET transaction 86
CEMT INQUIRE INTTRACE 219
CEMT INQUIRE transaction 218
CEMT SET transaction 412
INVALID
CEMT INQUIRE DSNAME 198
INVALID LDC AT XXXXXXXX
CMSG 530
INVALID OPND AT XXXXX
CMSG 530
INVALID TBL SFX AT XXXXX
CMSG 530
INVLD OPERATR ID AT XXXXX
CMSG 530
INVLD SEPARATOR AT XXXXX
CMSG 530
INVLD TERMINL ID XXXX
CMSG 530
invoking a transaction 3
invoking CEMT from an application program 132
IOERROR
CEMT INQUIRE UOWDSNFAIL 331
IOTYPE
CEMT INQUIRE TDQUEUE 299
IPCONN
CEMT DISCARD 136
CEMT INQUIRE UOWLINK 340
CEMT PERFORM STATISTICS 368
CEMT SET IPCONN 413
IPIC
CEMT INQUIRE UOWLINK 339
IRBATCH
CEMT INQUIRE transaction 224
IRC
CEMT INQUIRE CONNECTION 157
CEMT INQUIRE IRC 224
CEMT INQUIRE transaction 224
CEMT INQUIRE UOWLINK 339
CEMT SET transaction 417
IRC (interregion communication)
CEMT INQUIRE transaction 224
CEMT SET transaction 417
ISOLATE
CEMT INQUIRE TRANSACTION 310
ISOLATEST
CEMT INQUIRE TRANSACTION 310

J

JAVA
CEMT INQUIRE PROGRAM 260
JMODEL
CEMT DISCARD 136
CEMT INQUIRE JMODEL 226
CEMT transaction 225
JNDIPREFIX
CEMT INQUIRE CORBASERVER 166
JOURNALMODEL
CEDA 99
JOURNALNAME
CEMT DISCARD 136
CEMT INQUIRE JMODEL 226
CEMT INQUIRE JOURNALNAME 229
CEMT PERFORM STATISTICS 368
CEMT transaction 227
JVM
CEMT INQUIRE transaction 229
CEMT SET PROGRAM 433

JVMCLASS
CEMT INQUIRE PROGRAM 260
CEMT SET PROGRAM 433
JVMCOUNT
CEMT PERFORM JVMPOOL command 362
JVMPOOL
CEMT INQUIRE transaction 231
CEMT PERFORM STATISTICS 368
CEMT SET transaction 419
CEMT transaction 362
JVMProfile
CEMT SET PROGRAM 433
JVMPROFILE
CEMT INQUIRE PROGRAM 260
CEMT PERFORM JVMPOOL command 362
CEMT PERFORM STATISTICS 368
JVMPROGRAM
CEMT PERFORM STATISTICS 368

K

KEEPTIME
CEMT INQUIRE BRFACILITY 149
key, tab 129
KEYLENGTH
CEMT INQUIRE FILE 212
CEMT SET FILE 409
keyword, minimum abbreviation 4
KILL
CEKL SET TASK 119
CEMT SET IPCONN 415

L

LANGUAGE 475
CEMT INQUIRE PROGRAM 260
Language Environment 260
Language Environment runtime options (CLER) 505
LASTUSEDINT
CEMT INQUIRE TSQUEUE 322
CEMT SET TSQUEUE 458
LCKSTRUCFULL
CEMT INQUIRE UOWDSNFAIL 331
LDC (logical device code) 524
LDC NOT 2 CHARS AT XXXXX
CMSG 530
LENGTH
CEMT INQUIRE PROGRAM 260
CEMT INQUIRE TSQUEUE 322
LENV
CEMT INQUIRE PROGRAM 260
LIBRARY
CEMT DISCARD 136
CEMT INQUIRE PROGRAM 261
CEMT PERFORM STATISTICS 368
CEMT SET LIBRARY 420
LIBRARY option
INQUIRE LIBRARY command 234
SET LIBRARY command 421
LIBRARYDSN
CEMT INQUIRE PROGRAM 261
LINE
CEMT INQUIRE LINE 236
CEMT INQUIRE transaction 235
CEMT SET transaction 422
LINK
CEMT INQUIRE UOW 325
CEMT INQUIRE UOWLINK 339
LINKAUTH option
CEMT INQUIRE IPCONN command 221
LINKSYSNET
CEMT INQUIRE BRFACILITY 149
LINKSYSTEM
CEMT INQUIRE BRFACILITY 149
LIST
CEDA 99
list debugging profiles screen
application debugging profile manager
3270 interface 37
List profiles page
application debugging profile manager
web interface 19
LISTENERPORT option
INQUIRE WORKREQUEST command 352
Lists
of resource identifiers 128
LOAD
CEMT INQUIRE FILE 213
CEMT SET FILE 409
LOADTYPE
CEMT INQUIRE FILE 213
LOCAL
CEMT INQUIRE MONITOR 241
CEMT INQUIRE NETNAME 244
CEMT INQUIRE SYDUMPCODE 273
CEMT INQUIRE TERMINAL 305
CEMT INQUIRE TRDUMPCODE 314
CEMT SET SYDUMPCODE 439
CEMT SET TRDUMPCODE 457
LOCATION
CEMT INQUIRE TSMODEL 317
CEMT INQUIRE TSQUEUE 322
CEMT INQUIRE URIMAP 344
CEMT SET URIMAP 461
LOCKING
CEMT SET FILE 409
LOGDEFER
CEMT INQUIRE SYSTEM 277
CEMT SET SYSTEM 443
logical device code (LDC) 524
logical unit destination 524
LOSTLOCKS
CEMT INQUIRE DSNAME 196
LOW
CEMT INQUIRE DB2CONN 172, 178
CEMT SET DB2CONN 388
CEMT SET DB2ENTRY 393
LSRPOOL
CEDA 100
CEMT PERFORM STATISTICS 368
LTPNAME
CEMT INQUIRE PARTNER 248

LU61
CEMT INQUIRE CONNECTION 160
CEMT INQUIRE UOWLINK 339

M

MAIN
CEMT INQUIRE TSMODEL 317
CEMT INQUIRE TSQUEUE 322
manager, application debugging profile 17
MANUAL
CEBT SET TAKEOVER 87
Mappinglevel
CEMT INQUIRE WEBSERVICE 348
MAPSET
CEDA 100
CEMT INQUIRE PROGRAM 261
master terminal operator
duties 7
restriction 7
transaction, CEBT 79
transaction, CEKL 115
transaction, CEMT 123
types of terminal 7
master terminal, programmable interface to 132
MAXACTIVE
CEMT INQUIRE TCLASS 288
CEMT SET TCLASS 446
MAXDATALEN
CEMT SET TCPIPSERVICE 449
MAXIMUM
CEMT INQUIRE AUTOINSTALL 142
CEMT INQUIRE MODENAME 238
CEMT INQUIRE SYDUMPCODE 273
CEMT INQUIRE TRDUMPCODE 314
CEMT SET SYDUMPCODE 439
CEMT SET TRDUMPCODE 457
MAXITEMLEN
CEMT INQUIRE TSQUEUE 322
MAXJVMTCBS
CEMT INQUIRE DISPATCHER 182
CEMT SET DISPATCHER 397
MAXNUMRECS
CEMT INQUIRE FILE 213
CEMT SET FILE 409
MAXOPENTCBS
CEMT INQUIRE DISPATCHER 182
CEMT SET DISPATCHER 397
MAXQTIME
CEMT INQUIRE IPCONN 221
MAXREQS
CEMT SET AUTOINSTALL 372
MAXSOCKETS
CEMT INQUIRE TCPIP 289
CEMT SET TCPIP 447
MAXSSLTCBS
CEMT INQUIRE DISPATCHER 182
CEMT SET DISPATCHER 397
MAXTASKS
CEMT INQUIRE SYSTEM 277
CEMT SET SYSTEM 443
MAXXPTCBS
CEMT INQUIRE DISPATCHER 182
CEMT SET DISPATCHER 397
MEDIATYPE
CEMT INQUIRE URIMAP 344
MEMBER
CEMT INQUIRE TDQUEUE 299
MEMBERNAME
CEMT INQUIRE CONNECTION 159
MEMLIMIT
CEMT INQUIRE DSAS 190
CEMT INQUIRE SYSTEM 277
MESSAGE HAS BEEN ROUTED
CMSG 531
message-switching
continuation 513
errors 526
examples 533
message-switching response
indicator 526
response 526
message-switching transaction
multiple destinations 511
saved status 513
subsystems 514
terminal systems 514
transaction CMSG 511
messages
+DATE INVLD 2330 to 0030 533
ALL ROUTE ENTRIES DELETED 527
CANCELED - TEMP STRG ERR N 527
canceling 512
chaining 557
chaining in partition 9
CLEAR/CANCL or CONT INPUT/ROUTE/MSG 528
CONTINUE INPUT 528
CONTINUE INPUT OR MSG 528
CONTINUE MSG 528
CONTINUE ROUTE OPERAND 528
DATE ALREADY PASSED 528
DATE INVALID AT xxxxxxxx 529
DATE NOT xx/xx AT xxxxx 529
DATE TOO FAR IN FUTURE 529
delivery 514
ERRTERM INVALID AT xxxxx 529
ERRTERM INVLD/UNSUPP xxxx 529
HEADNG NOT YES/NO AT xxxx 529
ID OPERAND GT 62 CHARS 530
ID PARAM MISSING OR NO) 530
INVALID LDC AT xxxxxxxx 530
INVALID OPND AT xxxxx 530
INVALID TBL SFX AT xxxxx 530
INVLD OPERATR ID AT xxxxx 530
INVLD SEPARATOR AT xxxxx 530
INVLD TERMINL ID xxxx 530
IO OPERAND GT 62 CHARS 530
LDC NOT 2 CHARS AT xxxxx 530
MESSAGE HAS BEEN ROUTED 530
MESSAGE HAS BEEN SENT 563
MSG NOT SENT- NO RECEIVER 531
NEED OPCLASS OR ROUTE 531

messages *(continued)*
NO INPUT - REENTER 531
NO MESSAGE PROVIDED 531
NO MSG STRT QUOT AT xxxxx 531
NO TERM LIST TBL- xx 531
NO TRMID FOR LDC AT xxxxx 531
OPCLASS NOT 1-24 AT xxxxx 531
OPERATORS NOT SIGNDON NNN 531
OPRTR ID NOT SIGNDON /xxx 531
PROTECT OPND NOT 1-2 CHAR 532
purging 10
ROUTE INVALID AT xxxxx 532
sending 511
SPACE MUST FOLLOW TRANSID 532
TERMID GT 4 CHAR AT xxxxx 532
TERMID NOT +/- AT xxxxx 532
TERMINATED BY OPERATOR 532, 563
terminating in partition 10
termination 559
TIME ALREADY PASSED 532
TIME INVALID AT xxxxx 532
TIME NOT 4 CHARS AT xxxxx 533
TOO MANY TBL SFX AT xxxxx 533
UNBALANCD PARENS AT xxxxx 533
UNSUP OPRT TRMNL xxxxxxxx 533
UNSUP TERMINAL xxxxxxxx 533
ZERO LENGTH MSG AT " 533
messages and codes, display 507
MESSCHN
CSPG 557
MINITEMLEN
CEMT INQUIRE TSQUEUE 322
Minrunlevel
CEMT INQUIRE WEBSERVICE 349
minus sign following CEMT 123
MM/DD
CMSG 516, 518
MM/DD/YY
CMSG 516
MM/DD/YYYY
CMSG 518
MOD
CEMT INQUIRE TDQUEUE 298
MODE
CEMT INQUIRE PIPELINE 250
MODENAME
CEMT INQUIRE MODENAME 238
CEMT INQUIRE transaction 236
CEMT SET transaction 423
MODENAME, CEMT requests 423
MODIFY command 11
MODULE
CEMT INQUIRE REQUESTMODEL 264
MONITOR
CEMT INQUIRE MONITOR 240
CEMT INQUIRE transaction 238
CEMT PERFORM STATISTICS 368
CEMT SET transaction 424
monitoring facility transaction CEMN 121
MQCONN
CEMT PERFORM STATISTICS 368
MROBATCH
CEMT INQUIRE DISPATCHER 183
CEMT INQUIRE SYSTEM 277
CEMT SET DISPATCHER 397
CEMT SET SYSTEM 443
MSG
CMSG 519, 528
MSG NOT SENT - NO RECEIVER
CMSG 531
MSGCOPY
CSPG 560
MSGQUEUE1
CEMT INQUIRE DB2CONN 171
CEMT SET DB2CONN 387
MSGQUEUE2
CEMT INQUIRE DB2CONN 171
CEMT SET DB2CONN 387
MSGQUEUE3
CEMT INQUIRE DB2CONN 171
CEMT SET DB2CONN 387
MSGTERM
CSPG 559
MTOMNOXOPST
CEMT INQUIRE PIPELINE 250
MTOMST
CEMT INQUIRE PIPELINE 250
multiple console support 10
multiple destinations, message switching 511
MVS
CEMT INQUIRE JMODEL 227
CEMT INQUIRE JOURNALNAME 229

N

N906D
CEMT INQUIRE DB2CONN 173
NAME
CEMT INQUIRE DOCTEMPLATE 188
NAMEIND
CEMT INQUIRE TDQUEUE 299
NAMESPACE
CEMT INQUIRE BRFACILITY 149
NATLANG, system initialization parameter 475
NATURE
CEMT INQUIRE NETNAME 244
CEMT INQUIRE TERMINAL 304
NEED OPCLASS OR ROUTE
CMSG 531
NET
CEOT 468
NETID
CEMT PERFORM ENDAFFINITY 361
NETNAME
CEMT INQUIRE BRFACILITY 149
CEMT INQUIRE CONNECTION 159
CEMT INQUIRE NETNAME 244
CEMT INQUIRE PARTNER 248
CEMT INQUIRE TERMINAL 305
CEMT INQUIRE transaction 241
CEMT INQUIRE UOW 325
CEMT INQUIRE UOWDSNFAIL 330

NETNAME *(continued)*
CEMT PERFORM ENDAFFINITY 361
CEMT SET transaction 427
NETUOWID
CEMT INQUIRE ENQ 203
CEMT INQUIRE UOW 325
CEMT INQUIRE UOWENQ 335
CEMT INQUIRE UOWLINK 339
NETWORK
CEMT INQUIRE PARTNER 248
networks
CEMT INQUIRE transaction 241
CEMT SET transaction 427
NEWCOPY
CEMT INQUIRE PROGRAM 258
CEMT SET DOCTEMPLATE 399
CEMT SET PROGRAM 434
NEWPS 475
NEXT
CEBT INQUIRE AUXTRACE 80
CEBT SET AUXTRACE 84
CEMT INQUIRE AUXTRACE 144
CEMT SET AUXTRACE 372
NEXTTIME
CEMT INQUIRE STATISTICS 265, 267
CEMT SET STATISTICS 435
NO
CMSG 519, 520
NO INPUT - REENTER
CMSG 531
NO MESSAGE PROVIDED
CMSG 531
NO MSG STRT QUOT AT XXXXX
CMSG 531
NO TERM LIST TBL-XX
CMSG 531
NO TRMID FOR LDC AT XXXXX
CMSG 531
NOADDABLE
CEMT INQUIRE FILE 210
CEMT SET FILE 409
NOANALYZER
CEMT INQUIRE URIMAP 343
NOAPPLNAME
CEMT INQUIRE MONITOR 239
NOATI
CEMT INQUIRE NETNAME 243
CEMT INQUIRE TERMINAL 303
CEMT SET NETNAME 428
CEMT SET TERMINAL 453
CEOT 469
NOAUTO
CEMT SET CORBASERVER 383
NOAUTOSWITCH
CEBT INQUIRE DUMP 81
CEBT SET DUMP 85
CEMT INQUIRE DUMPDS 200
CEMT SET DUMPDS 405
NOBROWSE
CEMT INQUIRE FILE 210
CEMT SET FILE 409
NOCEDF
CEMT INQUIRE PROGRAM 258
CEMT SET PROGRAM 434
NOCMDPROT
CEMT INQUIRE SYSTEM 275
NOCOMPRESS
CEMT INQUIRE MONITOR 239
CEMT SET MONITOR 426
NOCONNECT
CEMT INQUIRE DB2CONN 173
CEMT SET DB2CONN 389
NOCONVERSE
CEMT INQUIRE MONITOR 240
CEMT SET MONITOR 426
NOCREATE
CEMT INQUIRE NETNAME 244
CEMT INQUIRE TERMINAL 304
CEMT SET NETNAME 429
CEMT SET TERMINAL 453
NODAE
CEMT INQUIRE SYDUMPCODE 273
CEMT SET SYDUMPCODE 439
NODEBUG
CEMT SET SYSTEM 443
NODELETE
CEMT INQUIRE FILE 211
CEMT SET FILE 409
NOEMPTYREQ
CEMT INQUIRE FILE 211
CEMT SET FILE 409
NOEXCEPT
CEMT INQUIRE MONITOR 240
CEMT SET MONITOR 426
NOEXCLUSIVE
CEMT INQUIRE FILE 212
CEMT SET FILE 410
NOFORCE
CEMT INQUIRE SYSTEM 277
CEMT SET SYSTEM 443
NOISOLATE
CEMT INQUIRE TRANSACTION 310
NOLOAD
CEMT INQUIRE FILE 213
CEMT SET FILE 410
NOLOSTLOCKS
CEMT INQUIRE DSNAME 196
NONCRITICAL value
INQUIRE TCPIPSERVICE 293
NONE
CEMT INQUIRE DB2CONN 168
CEMT INQUIRE DB2ENTRY 176
CEMT INQUIRE URIMAP 344
CEMT SET DB2CONN 384
CEMT SET DB2ENTRY 391
CEMT SET URIMAP 460
NONTERMREL
CEMT INQUIRE DB2CONN 171
CEMT SET DB2CONN 387
NOPERF
CEMT INQUIRE MONITOR 240
CEMT SET MONITOR 426

NOREAD
CEMT INQUIRE FILE 214
CEMT SET FILE 410
NORECOVDATA
CEMT INQUIRE CONNECTION 161
CEMT INQUIRE IPCONN 222
CEMT SET CONNECTION 378
CEMT SET IPCONN 415
NOREENTPROTECT
CEMT INQUIRE SYSTEM 278
NORELEASE
CEMT INQUIRE DB2CONN 171
CEMT SET DB2CONN 387
NORESRCE
CEMT INQUIRE MONITOR 240
CEMT SET MONITOR 426
NORESTART
CEMT PERFORM SHUTDOWN 365
NORESYNC
CEMT INQUIRE DB2CONN 172
CEMT SET DB2CONN 388
NORETAINED
CEMT INQUIRE DSNAME 198
NORMI
CEMT INQUIRE MONITOR 241
NOROLLBACK
CEMT INQUIRE DB2CONN 171
CEMT INQUIRE DB2ENTRY 177
NOSDTRAN
CEMT PERFORM SHUTDOWN 365
NOSECURITY
CEMT INQUIRE TSMODEL 317
NOSHUTDOWN
CEMT INQUIRE SYDUMPCODE 273
CEMT INQUIRE TRDUMPCODE 315
CEMT SET SYDUMPCODE 439
CEMT SET TRDUMPCODE 457
NOSWITCH
CEBT INQUIRE AUXTRACE 80
CEBT SET AUXTRACE 84
CEMT INQUIRE AUXTRACE 144
CEMT SET AUXTRACE 372
NOSYNCPOINT
CEMT INQUIRE MONITOR 241
CEMT SET MONITOR 426
NOSYSDUMP
CEMT INQUIRE SYDUMPCODE 274
CEMT INQUIRE TRDUMPCODE 315
CEMT SET SYDUMPCODE 439
CEMT SET SYSTEM 443
CEMT SET TRDUMPCODE 457
NOSYSLOG
CEMT INQUIRE STREAMNAME 270
NOTAPPLIC
CEMT INQUIRE DSNAME 195
CEMT INQUIRE FILE 214
CEMT INQUIRE UOWDSNFAIL 331
INQUIRE TCPIPSERVICE 293
notation, syntax 3
NOTCONNECTED
CEMT INQUIRE DB2CONN 170
NOTCONNECTED *(continued)*
CEMT SET DB2CONN 387
NOTDEFINED
CEMT INQUIRE PROGRAM 260
NOTDYNAMIC
CEMT INQUIRE PROGRAM 259
NOTEXTENDED
CEMT INQUIRE FILE 214
NOTPENDING
CEMT INQUIRE CONNECTION 159
CEMT INQUIRE IPCONN 222
CEMT SET CONNECTION 379
CEMT SET IPCONN 416
NOTPURGEABLE
CEMT INQUIRE TRANSACTION 311
CEMT SET TRANSACTION 455
NOTRANDUMP
CEMT INQUIRE TRDUMPCODE 315
CEMT SET TRDUMPCODE 457
NOTRECOVABLE
CEMT INQUIRE DSNAME 197
CEMT INQUIRE TSMODEL 317
CEMT INQUIRE TSQUEUE 323
NOTREQUIRED
CEMT INQUIRE PROGRAM 258
NOTRLS
CEMT INQUIRE FILE 214
CEMT INQUIRE UOWDSNFAIL 332
CEMT SET FILE 410
NOTROUTABLE
CEMT INQUIRE TRANSACTION 311
NOTSOS
CEMT INQUIRE DSAS 190, 191, 279
CEMT INQUIRE SYSTEM 279
NOTSUPPORTED
CEMT INQUIRE CONNECTION 158
NOTTABLE
CEMT INQUIRE FILE 215
CEMT SET FILE 410
NOTTI
CEMT INQUIRE NETNAME 247
CEMT INQUIRE TERMINAL 307
CEMT SET NETNAME 429
CEMT SET TERMINAL 453
CEOT 469
NOTWAIT
CEMT INQUIRE DB2CONN 174
CEMT INQUIRE DB2ENTRY 178
CEMT SET DB2CONN 390
CEMT SET DB2ENTRY 394
NOUPDATE
CEMT INQUIRE FILE 215
CEMT SET FILE 410
NOWAIT
CEMT INQUIRE TASK 285
CEMT INQUIRE TDQUEUE 298
CEMT INQUIRE TRANSACTION 310
NQNAME
CEMT INQUIRE CONNECTION 159
CEMT INQUIRE NETNAME 245
CEMT INQUIRE TERMINAL 305

NRS
CEMT INQUIRE CONNECTION 161
CEMT INQUIRE IPCONN 222
NUMBER
CEMT INQUIRE LINE 236
CEMT SET LINE 423
NUMDSNAMES option
INQUIRE LIBRARY command 234
NUMITEMS
CEMT INQUIRE TSQUEUE 323

O

OAPPLID option
INQUIRE WORKREQUEST command 352
OBJECT
CEMT INQUIRE DSNAME 197
OBTAINING
CEMT INQUIRE CONNECTION 158
CEMT INQUIRE IPCONN 220
OFF
CEBT SET SURVEILLANCE 87
CEDF 109
CEDX 113
CEMT INQUIRE MONITOR 241
CEMT INQUIRE STATISTICS 268
CEMT SET MONITOR 426
CEMT SET PROCESSTYPE 431
CEMT SET STATISTICS 437
CSFE SPOOLFE 550
CSFE ZCQTRACE 550
OK
CEMT INQUIRE STREAMNAME 270
CEMT INQUIRE UOWLINK 339
OLD
CEMT INQUIRE FILE 211
CEMT INQUIRE TDQUEUE 298
CEMT SET FILE 410
OLDCACHES
CEMT INQUIRE CLASSCACHE 153
ON
CEBT SET SURVEILLANCE 87
CEDF 110
CEDX 113
CEMT INQUIRE MONITOR 241
CEMT INQUIRE STATISTICS 268
CEMT SET MONITOR 427
CEMT SET STATISTICS 437
CSFE SPOOLFE 550
OPCLASS
CMSG 520
OPCLASS NOT 1-24 AT XXXXX
CMSG 531
OPEN
CEBT INQUIRE DUMP 81
CEBT SET DUMP 85
CEMT INQUIRE DUMPDS 200
CEMT INQUIRE FILE 213
CEMT INQUIRE IRC 225
CEMT INQUIRE RRMS 265
CEMT INQUIRE TDQUEUE 299
OPEN *(continued)*
CEMT INQUIRE VTAM 346
CEMT SET DUMPDS 405
CEMT SET FILE 410
CEMT SET IRC 418
CEMT SET VTAM 462
OPENAPI
CEMT INQUIRE PROGRAM 258
OPENERROR
CEMT INQUIRE UOWDSNFAIL 331
OPENSTATUS
CEMT INQUIRE DUMPDS 200
CEMT INQUIRE FILE 213
CEMT INQUIRE IRC 225
CEMT INQUIRE RRMS 265
CEMT INQUIRE TDQUEUE 299
CEMT INQUIRE VTAM 346
CEMT SET TCPIP 447
CEMT SET TCPIPSERVICE 449
operating system console as a terminal 10
OPERATION
CEMT INQUIRE REQUESTMODEL 264
operator security 5
OPERATORS NOT SIGNDON NNN
CMSG 531
OPID
CEMT INQUIRE DB2CONN 169
CEMT INQUIRE DB2ENTRY 176
CEMT SET DB2CONN 385
CEMT SET DB2ENTRY 392
CMSG 521
OPRTR ID NOT SIGNDON /XXX
CMSG 532
OSLEVEL
CEMT INQUIRE SYSTEM 277
OTASK option
INQUIRE WORKREQUEST command 352
OTRANSID option
INQUIRE WORKREQUEST command 353
OTSCOORD
CEMT INQUIRE UOW 327
OTSTID
CEMT INQUIRE UOW 326
OTSTIMEOUT
CEMT INQUIRE TRANSACTION 310
OUTPUT
CEMT INQUIRE TDQUEUE 299
OUTSERVICE
CEMT INQUIRE CONNECTION 162
CEMT INQUIRE IPCONN 223
CEMT INQUIRE NETNAME 246
CEMT INQUIRE TERMINAL 306
CEMT SET CONNECTION 379
CEMT SET IPCONN 416
CEMT SET LINE 423
CEMT SET NETNAME 429
CEMT SET TERMINAL 453
overtyping a display 128
OWNER
CEMT INQUIRE ENQ 203
CEMT INQUIRE UOWENQ 335

P

PA1 print key 9
PAG
  CEOT 468
page copying 9, 560
page retrieval
  and partitions 9, 556
  logical device 555
  session 553
  single keystroke 555
  transaction, CSPG 553
page retrieval function
  new PROFILE definition, DFHCICSP 5
PAGEABLE
  CEMT INQUIRE NETNAME 245
  CEMT INQUIRE TERMINAL 305
  CEMT SET NETNAME 429
  CEMT SET TERMINAL 454
  CEOT 469
PAGESTATUS
  CEMT INQUIRE NETNAME 245
  CEMT INQUIRE TERMINAL 305
paging command 553
PARAMETERS
  CSPG 557
partitions 556
partitions, BMS 8
PARTITIONSET
  CEDA 100
  CEMT INQUIRE PROGRAM 261
PARTNER
  CEDA 100
  CEMT DISCARD 137
  CEMT INQUIRE PARTNER 248
  CEMT INQUIRE transaction 247
PATH
  CEMT INQUIRE DSNAME 197
  CEMT INQUIRE URIMAP 343
PAUSE
  CEBT INQUIRE AUXTRACE 80
  CEBT SET AUXTRACE 84
  CEMT INQUIRE AUXTRACE 144
  CEMT SET AUXTRACE 373
PENDING
  CEMT INQUIRE CONNECTION 159
  CEMT INQUIRE IPCONN 222
PENDSTATUS
  CEMT INQUIRE CONNECTION 159
  CEMT INQUIRE IPCONN 222
PERF
  CEMT INQUIRE MONITOR 240
  CEMT SET MONITOR 427
PERFCLASS
  CEMT INQUIRE MONITOR 240
PERFORM, CEBT
  SHUTDOWN 88
  SNAP 88
  TAKEOVER 89
PERFORM, CEMT
  RESET 364
  SECURITY 364
PERFORM, CEMT *(continued)*
  SHUTDOWN 365
  STATISTICS 366
PERMANENT
  CEMT INQUIRE URIMAP 344
  CEMT SET URIMAP 461
PF (program function) key 126
PGCHAIN, system initialization parameter 553, 557
PGCOPY, system initialization parameter 553, 560
PGMINTERFACE
  CEMT INQUIRE WEBSERVICE 349
PGPURGE, system initialization parameter 553, 559
PGRET, system initialization parameter 553, 554
PHASEIN
  CEMT INQUIRE PROGRAM 259
  CEMT SET PROGRAM 434
PHASEOUT
  CEMT PERFORM CLASSCACHE command 355
  CEMT PERFORM JVMPOOL command 363
  CEMT SET JVMPOOL 420
PHASINGOUT
  CEMT INQUIRE CLASSCACHE 153
  CEMT INQUIRE JVMPOOL 232
PHASINGOUTST
  CEMT INQUIRE JVM 231
PIPELINE
  CEDA 100
  CEMT DISCARD 137
  CEMT INQUIRE URIMAP 342, 343
  CEMT INQUIRE WEBSERVICE 349
  CEMT PERFORM PIPELINE 364
  CEMT PERFORM STATISTICS 368
  CEMT SET PIPELINE 429
  CEMT transaction 363
PLAN
  CEMT INQUIRE DB2CONN 171
  CEMT INQUIRE DB2ENTRY 177
  CEMT SET DB2CONN 388
  CEMT SET DB2ENTRY 393
PLANEXITNAME
  CEMT INQUIRE DB2CONN 171
  CEMT INQUIRE DB2ENTRY 177
  CEMT SET DB2CONN 388
  CEMT SET DB2ENTRY 393
PLi
  CEMT INQUIRE PROGRAM 260
PLT
  CEMT PERFORM SHUTDOWN 365
plus sign in scrolling under CEMT 129
POOL
  CEMT SET DB2ENTRY 393
POOLNAME
  CEMT INQUIRE TSMODEL 317
  CEMT INQUIRE TSQUEUE 320
  CEMT SET TSQUEUE 458
PORT
  CEMT INQUIRE IPCONN 222
positional fields 128
PREFIX
  CEMT INQUIRE TSMODEL 317
  CMSG 520

PRFILE
CEMT INQUIRE TRANSACTION 310
PRGDLAY, system initialization parameter 518
PRI
CEOT 468
PRINT
CSFE 549
print (PA1) key 9
PRIORITY
CEMT INQUIRE DB2CONN 171
CEMT INQUIRE DB2ENTRY 178
CEMT INQUIRE NETNAME 245
CEMT INQUIRE TASK 285
CEMT INQUIRE TERMINAL 305
CEMT INQUIRE TRANSACTION 310
CEMT SET DB2CONN 388
CEMT SET DB2ENTRY 393
CEMT SET NETNAME 429
CEMT SET TASK 445
CEMT SET TERMINAL 454
CEMT SET TRANSACTION 455
PRIVATE
CEMT INQUIRE PROGRAM 262
CEMT SET PROGRAM 434
PROCESS
CEMT INQUIRE TASK 285
CEMT SET PROCESSTYPE 431
PROCESSTYPE
CEDA 100
CEMT DISCARD 137
CEMT INQUIRE TASK 285
CEMT SET transaction 431
PROFILE
CEDA 100
CEMT DISCARD 137
CEMT INQUIRE CLASSCACHE 153
CEMT INQUIRE JVM 231
CEMT INQUIRE PARTNER 248
CEMT INQUIRE PROFILE 254
CEMT INQUIRE transaction 253
CEMT PERFORM CLASSCACHE command 357
profile manager, application debugging 17
profile manager, debugging
web interface 17
configuring 17
profile, rebuild 364
PROGAUTO
CEMT PERFORM STATISTICS 369
PROGAUTOCTLG
CEMT INQUIRE SYSTEM 277
CEMT SET SYSTEM 441, 442
PROGAUTOEXIT
CEMT INQUIRE SYSTEM 278
CEMT SET SYSTEM 443
PROGAUTOINST
CEMT INQUIRE SYSTEM 278
CEMT SET SYSTEM 441
PROGRAM
CEDA 100
CEMT DISCARD 137
CEMT INQUIRE AUTOINSTALL 142
PROGRAM *(continued)*
CEMT INQUIRE PROGRAM 261
CEMT INQUIRE transaction 254
CEMT INQUIRE TRANSACTION 311
CEMT INQUIRE URIMAP 343
CEMT INQUIRE WEBSERVICE 349
CEMT PERFORM STATISTICS 369
CEMT SET AUTOINSTALL 372
CEMT SET PROGRAM 432
CEMT SET transaction 432
program function (PF) key 126
programmable interface to master terminal 132
PROGTYPE
CEMT INQUIRE PROGRAM 261
PROTECT
CMSG 520
PROTECT OPND NOT 1-2 CHAR
CMSG 532
PROTECTNUM
CEMT INQUIRE DB2ENTRY 178
CEMT SET DB2ENTRY 393
PROTOCOL
CEMT INQUIRE CONNECTION 160
CEMT INQUIRE UOWLINK 339
PS
CESN 475
PSDINTERVAL
CEMT INQUIRE VTAM 346
CEMT SET VTAM 462
pseudo-display fields 128
PTHREADS
CEMT INQUIRE DB2ENTRY 178
PUBLISH
CEMT PERFORM CORBASERVER 358
CEMT PERFORM DJAR 360
PURGE
CEKL SET TASK 119
CEMT INQUIRE CONNECTION 160
CEMT INQUIRE NETNAME 245
CEMT INQUIRE TASK 285
CEMT INQUIRE TERMINAL 305
CEMT PERFORM CLASSCACHE command 355
CEMT PERFORM JVMPOOL command 363
CEMT SET CONNECTION 380
CEMT SET IPCONN 416
CEMT SET JVMPOOL 420
CEMT SET LINE 423
CEMT SET NETNAME 429
CEMT SET TASK 445
CEMT SET TERMINAL 454
CEMT SET WORKREQUEST 466
INQUIRE WORKREQUEST command 353
PURGEABILITY
CEMT INQUIRE TRANSACTION 311
PURGEABLE
CEMT INQUIRE TRANSACTION 311
CEMT SET TRANSACTION 455
PURGECYCLEM
CEMT INQUIRE DB2CONN 172
CEMT SET DB2CONN 388

PURGECYCLES
CEMT INQUIRE DB2CONN 172
CEMT SET DB2CONN 388
PURGETHRESH
CEMT INQUIRE TCLASS 288
CEMT SET TCLASS 446
PURGETYPE
CEMT INQUIRE CONNECTION 160
CEMT INQUIRE NETNAME 245
CEMT INQUIRE TASK 285
CEMT INQUIRE TERMINAL 305
PURGETYPE option
INQUIRE WORKREQUEST command 353

Q

QD
CEMT INQUIRE TASK 286
QLEN
CEMT INQUIRE ENQ 203
CEMT INQUIRE UOWENQ 335
QR
CEMT INQUIRE TASK 286
QUALIFIER
CEMT INQUIRE ENQ 203
CEMT INQUIRE UOWENQ 335
QUASIRENT
CEMT INQUIRE PROGRAM 258
question mark preceding CEMT 123
QUEUE
CEMT INQUIRE TDQUEUE 299
CEMT SET transaction 435
QUEUED
CEMT INQUIRE TCLASS 288
QUEUELIMIT
CEMT INQUIRE IPCONN 222
queues, CEMT requests 295, 435, 446
QUIESCED
CEMT INQUIRE DSNAME 197
CEMT SET DSNAME 402
QUIESCESTATE
CEMT INQUIRE DSNAME 197

R

RACF resource profile 364
RANKING option
INQUIRE LIBRARY command 235
SET LIBRARY command 422
RBATYPE
CEMT INQUIRE FILE 214
RCONNECTION
CEMT INQUIRE CONNECTION 157
RDBACK
CEMT INQUIRE TDQUEUE 299
RDO (resource definition online) 95
RDSASIZE
CEMT INQUIRE DSAS 190
READ
CEMT INQUIRE FILE 214
CEMT SET FILE 410
READSTATUS
CEMT INQUIRE FILE 214
REASON
CEMT INQUIRE UOWDSNFAIL 330
REBUILD
CEMT PERFORM SECURITY 365
CEMT SET DB2CONN 389
rebuild resource profiles 364
RECEIVECOUNT
CEMT INQUIRE IPCONN 222
RECONNECT
CEMT INQUIRE DB2CONN 173
CEMT SET DB2CONN 389
RECORD
CEMT INQUIRE STATISTICS 268
CEMT PERFORM STATISTICS 369
RECORDING
CEMT INQUIRE STATISTICS 268
RECORDNOW
CEMT SET STATISTICS 437
RECORDSIZE
CEMT INQUIRE FILE 214
CEMT SET FILE 410
RECOVDATA
CEMT INQUIRE CONNECTION 161
CEMT INQUIRE IPCONN 222
RECOVERABLE
CEMT INQUIRE DSNAME 198
CEMT INQUIRE TSMODEL 317
CEMT INQUIRE TSQUEUE 323
RECOVERED
CEMT INQUIRE DSNAME 194
CEMT SET DSNAME 402
RECOVERLOCKS
CEMT INQUIRE DSNAME 197
RECOVERY
CEMT PERFORM STATISTICS 369
RECOVSTATUS
CEMT INQUIRE CONNECTION 161
CEMT INQUIRE DSNAME 197
CEMT INQUIRE IPCONN 222
CEMT INQUIRE TSMODEL 317
CEMT INQUIRE TSQUEUE 323
REDIRECTTYPE
CEMT INQUIRE URIMAP 344
REENTPROTECT
CEMT INQUIRE SYSTEM 278
REGERROR
INQUIRE TCPIPSERVICE 293
REGISTERED
CEMT INQUIRE VTAM 345
INQUIRE TCPIPSERVICE 293
REJECT
CEMT INQUIRE TDQUEUE 299
RELATED
CEMT INQUIRE SYDUMPCODE 273
CEMT INQUIRE TRDUMPCODE 314
CEMT SET SYDUMPCODE 439
CEMT SET TRDUMPCODE 457
RELATION
CEMT INQUIRE ENQ 203

RELATION *(continued)*
CEMT INQUIRE UOWENQ 335
RELEASE
CEMT INQUIRE DB2CONN 171
CEMT INQUIRE SYSTEM 278
CEMT SET DB2CONN 387
RELEASED
CEMT INQUIRE CONNECTION 158
CEMT INQUIRE IPCONN 220
CEMT INQUIRE NETNAME 246
CEMT INQUIRE TERMINAL 306
CEMT SET BRFACILITY 374
CEMT SET CONNECTION 380
CEMT SET IPCONN 416
CEMT SET NETNAME 429
CEMT SET TERMINAL 454
RELOAD
CEMT PERFORM CLASSCACHE command 356
REMLOSTLOCKS
CEMT INQUIRE DSNAME 196
REMOTE
CEMT INQUIRE FILE 210
CEMT INQUIRE TDQUEUE 300
REMOTEPREFIX
CEMT INQUIRE TSMODEL 317
REMOTESYSNET
CEMT INQUIRE BRFACILITY 149
REMOTESYSTEM
CEMT INQUIRE BRFACILITY 149
CEMT INQUIRE CONNECTION 161
CEMT INQUIRE NETNAME 245
CEMT INQUIRE PROGRAM 261
CEMT INQUIRE TERMINAL 305
CEMT INQUIRE TSMODEL 317
REMOVE
CEMT INQUIRE DSNAME 193
CEMT INQUIRE SYDUMPCODE 272
CEMT INQUIRE TRDUMPCODE 314
CEMT SET DSNAME 402
CEMT SET SYDUMPCODE 440
CEMT SET TRDUMPCODE 457
REPLY command 12
REQUESTID option
INQUIRE WORKREQUEST command 353
REQUESTMODEL
CEDA 100
CEMT DISCARD 137
CEMT PERFORM STATISTICS 369
CEMT transaction 262
REQUIRED
CEMT INQUIRE PROGRAM 258
RESCOUNT
CEMT INQUIRE PROGRAM 261
RESET
CEMT INQUIRE JOURNALNAME 228
CEMT INQUIRE STATISTICS 268
CEMT INQUIRE SYDUMPCODE 272
CEMT INQUIRE TRDUMPCODE 314
CEMT PERFORM RESET 364
CEMT SET JOURNALNAME 419
CEMT SET SYDUMPCODE 440
RESET *(continued)*
CEMT SET TRDUMPCODE 457
RESETLOCKS
CEMT INQUIRE DSNAME 194
CEMT SET DSNAME 403
RESETNOW
CEMT PERFORM STATISTICS 369
CEMT SET STATISTICS 437
RESOURCE
CEMT INQUIRE ENQ 203
CEMT INQUIRE UOWENQ 335
resource definition online transaction, CEDA 95
resource definitions
profile, DFHCICSP 4
resource groups
Subdefining 128
resource identifiers
lists of 128
RESPWAIT
CEMT INQUIRE PIPELINE 250
CEMT SET PIPELINE 430
RESRCE
CEMT INQUIRE MONITOR 240
CEMT SET MONITOR 427
RESRCECLASS
CEMT INQUIRE MONITOR 240
RESYNC
CEMT INQUIRE CONNECTION 162
CEMT INQUIRE DB2CONN 172
CEMT SET CONNECTION 380
CEMT SET DB2CONN 388
CEMT SET IPCONN 416
RESYNCMEMBER
CEMT INQUIRE DB2CONN 172
CEMT SET DB2CONN 388
RESYNCSTATUS
CEMT INQUIRE UOWLINK 339
RETAINED
CEMT INQUIRE DSNAME 198
CEMT INQUIRE ENQ 203
CEMT INQUIRE UOWENQ 336
RETLOCKS
CEMT INQUIRE DSNAME 198
RETRACT
CEMT PERFORM CORBASERVER 358
CEMT PERFORM DJAR 360
retrieve a page 9, 553
RETRY
CEMT INQUIRE DSNAME 195
CEMT SET DSNAME 403
REUSEST
CEMT INQUIRE CLASSCACHE 151
CEMT INQUIRE JVM 231
RLEN
CEMT INQUIRE ENQ 203
CEMT INQUIRE UOWENQ 336
rls
CEMT INQUIRE UOWDSNFAIL 332
RLS
CEMT INQUIRE FILE 214
CEMT SET FILE 410

RLSACCESS
CEMT INQUIRE FILE 214
CEMT INQUIRE UOWDSNFAIL 332
RLSGONE
CEMT INQUIRE UOWDSNFAIL 331
RLSSERVER
CEMT INQUIRE UOW 327
CEMT INQUIRE UOWDSNFAIL 330
RMI
CEMT INQUIRE MONITOR 241
CEMT INQUIRE UOWLINK 340
RMIQFY
CEMT INQUIRE UOWLINK 339
RMIST
CEMT INQUIRE MONITOR 241
RNAME 162
CEMT INQUIRE NETNAME 245
CEMT INQUIRE TERMINAL 306
RNETNAME 162
CEMT INQUIRE NETNAME 246
CEMT INQUIRE TERMINAL 306
ROLE
CEMT INQUIRE UOWLINK 339
ROLLBACK
CEMT INQUIRE DB2CONN 171
CEMT INQUIRE DB2ENTRY 177
ROUTABLE
CEMT INQUIRE TRANSACTION 311
ROUTE
CMSG 520, 528
ROUTE INVALID AT XXXXX
CMSG 532
routed pages 561
routing transaction, CRTE 547
ROUTSTATUS
CEMT INQUIRE TRANSACTION 311
RRCOMMITFAIL
CEMT INQUIRE UOWDSNFAIL 332
RRINDOUBT
CEMT INQUIRE UOWDSNFAIL 332
RRMS
CEMT INQUIRE transaction 265
CEMT INQUIRE UOW 327
CEMT INQUIRE UOWLINK 339
RSESSION
CEMT INQUIRE NETNAME 244
CEMT INQUIRE TERMINAL 304
RTERMINAL
CEMT INQUIRE NETNAME 244
CEMT INQUIRE TERMINAL 304
RUNAWAY
CEMT INQUIRE DISPATCHER 183
CEMT INQUIRE SYSTEM 278
CEMT SET DISPATCHER 397
CEMT SET SYSTEM 443
RUNNING
CEKL INQUIRE TASK 116
CEMT INQUIRE TASK 285
running CIND 500
RUNSTATUS
CEMT INQUIRE TASK 285
RUNTIME
CEMT INQUIRE PROGRAM 261

S

S
CEMT INQUIRE TASK 286
saved status, message switching 513
SCAN
CEMT PERFORM CORBASERVER 358
CEMT PERFORM PIPELINE 364
SCANDELAY
CEMT INQUIRE DISPATCHER 183
CEMT INQUIRE SYSTEM 279
CEMT SET DISPATCHER 398
CEMT SET SYSTEM 444
SCHEME
CEMT INQUIRE URIMAP 343
scrolling a display 129
SD
CEMT INQUIRE TASK 286
SDSASIZE
CEMT INQUIRE DSAS 190
SDTRAN
CEMT INQUIRE SYSTEM 279
CEMT PERFORM SHUTDOWN 365
SEARCHPOS option
INQUIRE LIBRARY command 235
SECURITY 364
CEMT INQUIRE DB2CONN 172
CEMT INQUIRE TSMODEL 317
CEMT SET DB2CONN 388
security checking 103, 132
security, operator 5
SECURITYNAME
CEMT INQUIRE IPCONN 223
SECURITYST
CEMT INQUIRE TSMODEL 317
SEND
CMSG 523
SENDCOUNT
CEMT INQUIRE IPCONN 223
SENDMTOMST
CEMT INQUIRE PIPELINE 250
sequential terminal 515
SERVER
CEMT INQUIRE URIMAP 342
SERVSTATUS
CEMT INQUIRE CONNECTION 162
CEMT INQUIRE IPCONN 223
CEMT INQUIRE LINE 236
CEMT INQUIRE NETNAME 246
CEMT INQUIRE TERMINAL 306
SESSBEANTIME
CEMT INQUIRE CORBASERVER 166
CEMT SET CORBASERVER 383
SESSION
CEMT INQUIRE NETNAME 244
CEMT INQUIRE TERMINAL 304
SESSIONID
CEDF 110

SESSIONS
CEDA 100
set LE debugging display device screen
application debugging profile manager
3270 interface 54
Set LE display device page
application debugging profile manager
web interface 35
setting the display device
with the 3270 interface 53
with the web interface 34
SHARE
CEMT INQUIRE FILE 211
CEMT SET FILE 410
SHARED
CEMT INQUIRE PROGRAM 262
CEMT SET PROGRAM 435
SHARESTATUS
CEMT INQUIRE PROGRAM 262
SHELF
CEMT INQUIRE CORBASERVER 166
CEMT INQUIRE PIPELINE 251
shipped terminal definitions
deletion of
CEMT DJAR command 359
CEMT INQUIRE DELETSHIPPED 180
CEMT PERFORM CORBASERVER command 357
CEMT PERFORM DELETSHIPPED command 359
CEMT SET DELETSHIPPED command 395
SHR
CEMT INQUIRE TDQUEUE 298
SHUNTED
CEMT INQUIRE UOW 328
SHUTDOWN
CEMT INQUIRE SYDUMPCODE 273
CEMT INQUIRE TRDUMPCODE 315
CEMT SET SYDUMPCODE 440
CEMT SET TRDUMPCODE 457
shutdown, CICS
CEBT 88
CEMT 365
SHUTOPTION
CEMT INQUIRE SYDUMPCODE 273
CEMT INQUIRE TRDUMPCODE 315
SIGN
CEMT INQUIRE DB2ENTRY 176
CEMT SET DB2CONN 385
CEMT SET DB2ENTRY 392
sign-off, CESF
fixed-format terminal 471
SIGNID
CEMT INQUIRE DB2CONN 169, 172
CEMT SET DB2CONN 389
signon, CESN
3270 screen 474
non-3270 screen 475
single keystroke retrieval (SKR) 555
SIZE
CEMT INQUIRE DOCTEMPLATE 188
SKR (single keystroke retrieval) 555
SKRxxxx, system initialization parameter 555
SMF
CEMT INQUIRE JMODEL 227
CEMT INQUIRE JOURNALNAME 229
SNAP
CEMT PERFORM DUMP 361
CEMT transaction 360
SNAP command
CEBT transaction 88
SOAP
CEMT SET WORKREQUEST 465
SOAPlevel
CEMT INQUIRE PIPELINE 251
SOS
CEMT INQUIRE DSAS 190, 191, 279
CEMT INQUIRE SYSTEM 279
SOSABOVEBAR
CEMT INQUIRE DSAS 190, 279
SOSABOVELINE
CEMT INQUIRE DSAS 190
CEMT INQUIRE SYSTEM 279
SOSBELOWLINE
CEMT INQUIRE DSAS 191
CEMT INQUIRE SYSTEM 279
SPACE MUST FOLLOW TRANSID
CMSG 532
SPOOLFE
CSFE 550
spooling interface trace 550
SQLCODE
CEMT INQUIRE DB2CONN 170
CEMT INQUIRE DB2ENTRY 177
CEMT SET DB2CONN 386
CEMT SET DB2ENTRY 393
SSLTYPE option
CEMT INQUIRE IPCONN command 223
SSLUNAUTH
CEMT INQUIRE CORBASERVER 167
STACK option
INQUIRE WORKREQUEST command 353
STANDBYMODE
CEMT INQUIRE DB2CONN 173
CEMT SET DB2CONN 389
START
CEBT INQUIRE AUXTRACE 80
CEBT INQUIRE GTFTRACE 82
CEBT INQUIRE INTTRACE 82
CEBT SET AUXTRACE 84
CEBT SET GTFTRACE 86
CEBT SET INTTRACE 86
CEMT INQUIRE AUXTRACE 144
CEMT INQUIRE GTFTRACE 216
CEMT INQUIRE INTTRACE 219
CEMT PERFORM CLASSCACHE command 356
CEMT PERFORM JVMPOOL 363
CEMT SET AUXTRACE 373
CEMT SET GTFTRACE 411
CEMT SET INTTRACE 413
STARTCODE
CEMT INQUIRE TASK 286

STARTING
CEMT INQUIRE UOWLINK 339
starting a transaction 3
STATE
CEMT INQUIRE DJAR 185
CEMT INQUIRE ENQ 203
CEMT INQUIRE UOWENQ 336
STATIC
CEMT INQUIRE DSNAME 195
STATISTICS
CEMT INQUIRE STATISTICS 268
CEMT INQUIRE transaction 265
CEMT PERFORM 366
CEMT SET transaction 435
STATS
CEMT PERFORM STATISTICS 369
STATSQUEUE
CEMT INQUIRE DB2CONN 173
CEMT SET DB2CONN 389
status
hardcopy terminal 561
indicator 526
STATUS
CEMT INQUIRE AUXTRACE 144
CEMT INQUIRE CLASSCACHE 154
CEMT INQUIRE ENQMODEL 206
CEMT INQUIRE GTFTRACE 216, 219
CEMT INQUIRE JOURNALNAME 229
CEMT INQUIRE JVMPOOL 232
CEMT INQUIRE MONITOR 241
CEMT INQUIRE PROGRAM 262
CEMT INQUIRE STREAMNAME 270
CEMT INQUIRE TRANSACTION 311
STOP
CEBT INQUIRE AUXTRACE 80
CEBT SET AUXTRACE 84
CEBT SET GTFTRACE 86
CEBT SET INTTRACE 86
CEMT INQUIRE AUXTRACE 144
CEMT INQUIRE GTFTRACE 216
CEMT INQUIRE INTTRACE 219
CEMT SET AUXTRACE 373
CEMT SET GTFTRACE 411
CEMT SET INTTRACE 413
STOPPED
CEBT INQUIRE GTFTRACE 82
CEBT INQUIRE INTTRACE 82
STORAGE
CEMT PERFORM STATISTICS 369
storage subpool 551
STOREPROTECT
CEMT INQUIRE SYSTEM 279
STREAMNAME
CEMT INQUIRE JMODEL 227
CEMT INQUIRE JOURNALNAME 229
CEMT INQUIRE STREAMNAME 270
CEMT PERFORM STATISTICS 369
CEMT transaction 269
Subdefining
resource groups 128
SUBORDINATE
CEMT INQUIRE UOWLINK 339
subsystem message-switching 514
SUBTASKS
CEMT INQUIRE DISPATCHER 183
supervisory terminal operator
duties 6
identifying terminals 6
SUPRID option 6
transaction CEST 477
SURVEILLANCE
CEBT transaction 83, 87
SUSPENDED
CEKL INQUIRE TASK 116
CEMT INQUIRE TASK 286
SWITCH
CEBT INQUIRE AUXTRACE 80
CEBT INQUIRE DUMP 81
CEBT SET AUXTRACE 84
CEBT SET DUMP 85
CEMT INQUIRE AUXTRACE 144
CEMT INQUIRE DUMPDS 200
CEMT SET AUXTRACE 373
CEMT SET DUMPDS 405
SWITCHACTION
CEMT INQUIRE AUXTRACE 144
SWITCHSTATUS
CEMT INQUIRE AUXTRACE 144
CEMT INQUIRE DUMPDS 200
SYDUMPCODE
CEMT INQUIRE SYDUMPCODE 273
CEMT SET transaction 437
CEMT transaction 270
SYNCPOINT
CEMT INQUIRE MONITOR 241
CEMT SET MONITOR 427
SYNCPOINTST
CEMT INQUIRE MONITOR 241
syntax notation 3
SYSDUMP
CEMT INQUIRE SYDUMPCODE 274
CEMT INQUIRE TRDUMPCODE 315
CEMT PERFORM STATISTICS 369
CEMT SET SYDUMPCODE 440
CEMT SET SYSTEM 444
CEMT SET TRDUMPCODE 457
SYSDUMPING
CEMT INQUIRE SYDUMPCODE 274
CEMT INQUIRE TRDUMPCODE 315
SYSID
CEDF 110
CEMT INQUIRE TSQUEUE 320
CEMT INQUIRE UOW 326
CEMT INQUIRE UOWDSNFAIL 332
CEMT INQUIRE UOWLINK 340
CEMT SET TSQUEUE 458
CRTE transaction 547
SYSLOG
CEMT INQUIRE STREAMNAME 270
SYSTEM
CEMT INQUIRE transaction 274

SYSTEM *(continued)*
CEMT SET transaction 440
system spooling interface trace 550
SYSTEMLOG
CEMT INQUIRE STREAMNAME 270

T

tab key 129
TABLE
CEMT INQUIRE FILE 215
TABLEMGR
CEMT PERFORM STATISTICS 369
TABLENAME
CEMT INQUIRE FILE 215
CEMT SET FILE 410
TAKEOVER
CEBT 83
CEBT transaction 87, 89
CEMT PERFORM SHUTDOWN 366
task
CEKL INQUIRE TASK 116
Task
CEKL SET TASK 119
TASK
CEKL SET transaction 118
CEMT INQUIRE DB2CONN 168
CEMT INQUIRE DB2ENTRY 175
CEMT INQUIRE EXCI 207
CEMT INQUIRE JVM 231
CEMT INQUIRE NETNAME 246
CEMT INQUIRE TASK 284, 286
CEMT INQUIRE TERMINAL 306
CEMT SET DB2CONN 384
CEMT SET DB2ENTRY 391
CEMT SET transaction 444
TASK command
CEKL INQUIRE transaction 115
CEMT INQUIRE transaction 280
TASK option
INQUIRE WORKREQUEST command 353
TASKDATAKEY
CEMT INQUIRE TRANSACTION 311
TASKDATALOC
CEMT INQUIRE TRANSACTION 312
TASKID
CEMT INQUIRE BRFACILITY 149
CEMT INQUIRE ENQ 203
CEMT INQUIRE UOW 326
CEMT INQUIRE UOWENQ 336
tasks
CEKL INQUIRE requests 115
CEKL SET requests 118
CEMT INQUIRE requests 280
CEMT SET requests 444
TCB
CEMT INQUIRE TASK 286
TCBLIMIT
CEMT INQUIRE DB2CONN 173
CEMT SET DB2CONN 389
TCBs
CEMT INQUIRE DB2CONN 173
TCLASS
CEMT DISCARD 137
CEMT INQUIRE TASK 281
CEMT INQUIRE TCLASS 288
CEMT INQUIRE transaction 287
CEMT INQUIRE TRANSACTION 312
CEMT PERFORM STATISTICS 369
CEMT SET transaction 445
CEMT SET TRANSACTION 455
TCP/IP services, CEMT requests 288, 290
TCPIP
CEMT INQUIRE transaction 288
CEMT PERFORM STATISTICS 369
CEMT SET transaction 446
TCPIP, CEMT requests 448
TCPipservice
CEMT PERFORM STATISTICS 369
TCPIPSERVICE
CEDA 100
CEMT DISCARD 137
CEMT INQUIRE HOST 217
CEMT INQUIRE IPCONN 223
CEMT INQUIRE transaction 290
CEMT INQUIRE URIMAP 343
CEMT SET transaction 448
TDQ
CEMT INQUIRE ENQ 204
CEMT INQUIRE UOWENQ 336
TDQUEUE
CEDA 100
CEMT DISCARD 137
CEMT INQUIRE TDQUEUE 300
CEMT INQUIRE transaction 295
CEMT PERFORM STATISTICS 369
CEMT SET transaction 449
TEMPLATENAME
CEMT INQUIRE DOCTEMPLATE 188
CEMT INQUIRE URIMAP 343
TEMPLATETYPE
CEMT INQUIRE DOCTEMPLATE 188
TEMPORARY
CEMT INQUIRE URIMAP 344
CEMT SET URIMAP 460
temporary storage browse transaction, CEBR 75
TER
CEOT 468
TERM
CEMT INQUIRE DB2CONN 169
CEMT INQUIRE DB2ENTRY 176
CEMT INQUIRE TASK 284
CEMT SET DB2CONN 385
CEMT SET DB2ENTRY 392
TERMID
CEDF 110
CEMT INQUIRE BRFACILITY 150
CEMT INQUIRE TDQUEUE 300
CEMT INQUIRE UOW 326
CMSG 521
CSFE ZCQTRACE 550

TERMID GT 4 CHAR AT XXXXX
CMSG 532
TERMID NOT +/- AT xxxxx
CMSG 532
TERMID/OPID
CMSG 521
TERMID*LDC
CMSG 525
TERMINAL
CEDA 100
CEMT DISCARD 137
CEMT INQUIRE NETNAME 246
CEMT INQUIRE TERMINAL 306
CEMT INQUIRE transaction 300
CEMT PERFORM STATISTICS 369
CEMT SET transaction 451
terminal and system test transaction, CSFE 549
terminal builder trace 550
terminal list table (TLT) 6
terminal operator
duties 6
master 7
paging 553
supervisory 6
transactions 5
write to console operator (CWTO) 563
terminals
3270 displays 515
3767 keyboard/printer 515
autopaging 561
CEMT INQUIRE requests 300
CEMT SET requests 449, 451
console as a CICS terminal 10
fixed-format 471
identifying 6
message-switching transaction 514
paging 9, 553
sequential 515
status 467
test 549
terminate a logical message 10, 559
TERMINATED BY OPERATOR
CMSG 532
TERMLIST
CMSG 522
TERMLIST*LDC
CMSG 525
TERMSTATUS
CEMT INQUIRE BRFACILITY 150
CEMT INQUIRE NETNAME 246
CEMT INQUIRE TERMINAL 306
testing a terminal 549
THREADERROR
CEMT INQUIRE DB2CONN 173
THREADLIMIT
CEMT INQUIRE DB2CONN 174
CEMT INQUIRE DB2ENTRY 178
CEMT SET DB2CONN 389
CEMT SET DB2ENTRY 393
THREADS
CEMT INQUIRE DB2CONN 174
THREADS *(continued)*
CEMT INQUIRE DB2ENTRY 178
THREADSAFE
CEMT INQUIRE PROGRAM 258
THREADWAIT
CEMT INQUIRE DB2CONN 174
CEMT INQUIRE DB2ENTRY 178
CEMT SET DB2CONN 389
CEMT SET DB2ENTRY 394
TIME
CEMT INQUIRE DISPATCHER 183
CEMT INQUIRE MONITOR 241
CEMT INQUIRE SYSTEM 280
CEMT SET DISPATCHER 398
CEMT SET SYSTEM 444
CMSG 523
TIME ALREADY PASSED
CMSG 532
TIME INVALID AT XXXXX
CMSG 532
TIME NOT 4 CHARS AT XXXXX
CMSG 533
time-of-day resetting 364
TIMEOUTINT
CEMT INQUIRE WEB 347
CEMT SET WEB 463
TIMESTAMP
CEMT INQUIRE DJAR 185
TIMESTARTED
CEMT INQUIRE CLASSCACHE 154
TITLE
CEMT PERFORM DUMP 361
TLT (terminal list table) 6
TO
CEDA 100
CEMT INQUIRE TASK 286
TOO MANY TBL SFX AT XXXXX
CMSG 533
TOTAL
CEMT INQUIRE JVMPOOL 232
TOTALJVMS
CEMT INQUIRE CLASSCACHE 154
TP
CEMT INQUIRE TASK 286
TPNAME
CEMT INQUIRE PARTNER 248
TPOOL
CEMT INQUIRE DB2ENTRY 178
CEMT SET DB2ENTRY 394
TRA
CEOT 468
trace control transaction 479
trace program exit routine 552
TRANClass
CEKL INQUIRE TASK 116
TRANCLASS
CEDA 100
TRANDUMP
CEMT INQUIRE TRDUMPCODE 315
CEMT SET TRDUMPCODE 457

TRANDUMPING
CEMT INQUIRE TRDUMPCODE 315
TRANID
CEDX 113
CEMT INQUIRE TASK 286
CEMT INQUIRE TDQUEUE 300
TRANISOLATE
CEMT INQUIRE SYSTEM 280
TRANSACTION
CEDA 101
CEMT DISCARD 138
CEMT INQUIRE BRFACILITY 150
CEMT INQUIRE NETNAME 247
CEMT INQUIRE TERMINAL 307
CEMT INQUIRE transaction 307
CEMT INQUIRE TRANSACTION 312
CEMT INQUIRE URIMAP 343
CEMT PERFORM STATISTICS 369
CEMT SET transaction 454
transaction routing, CRTE 547
transaction, BTS browser 57
transaction, in-doubt test tool 499
transaction, load module map 503
transaction, monitoring facility 121
transaction, trace control 479
transactions
all (list) 567
canceling 3
CBAM 57
CCRL 59
CEBT 79
CECI 91
CECS 91, 93
CEDA 95
CEDB 95
CEDC 95
CEDF 109
CEKL 115
CEMT 123
CEOT 467
CESN 473
CEST 477
CMSG 511
CREA 535
CREC 535
CRTE 547
CSFE 549
CSPG 553
CWTO 563
DSNC 565
identification codes 6
initiating from console 11
invocation of 3
security key 6
transactions with operator interface 567
transactions with operator interface
CADP 37
CBAM 57
CDBC 61
CDBI 65
CDBM 67
transactions with operator interface *(continued)*
CEMN 121
CESF 471
CETR 479
CIDP 497
CLER 505
CMAC 507
TRANSID
CEKL INQUIRE TASK 116
CEMT INQUIRE DB2TRAN 179
CEMT INQUIRE ENQ 203
CEMT INQUIRE REQUESTMODEL 264
CEMT INQUIRE TSQUEUE 323
CEMT INQUIRE UOW 326
CEMT INQUIRE UOWENQ 336
CEMT SET DB2TRAN 395
TRANSID option
INQUIRE WORKREQUEST command 353
TRDUMP
CEMT PERFORM STATISTICS 369
TRDUMPCODE
CEMT INQUIRE transaction 312
CEMT INQUIRE TRDUMPCODE 315
CEMT SET transaction 456
TRIGGERLEVEL
CEMT INQUIRE TDQUEUE 300
TRPROF
CEMT INQUIRE TRANSACTION 312
CRTE transaction 547
TSMODEL
CEDA 101
CEMT DISCARD 138
CEMT INQUIRE transaction 315
CEMT INQUIRE TSMODEL 317
TSO console as a CICS terminal 13
TSPOOL
CEMT INQUIRE transaction 318
CEMT INQUIRE TSPOOL 319
TSQ
CEMT INQUIRE ENQ 204
CEMT INQUIRE UOWENQ 336
TSQNAME
CEMT INQUIRE transaction 319
TSQUEUE
CEMT INQUIRE transaction 319
CEMT INQUIRE TSQUEUE 323
CEMT PERFORM STATISTICS 369
CEMT SET transaction 457
CEMT SET TSQUEUE 458
TSQUEUELIMIT
CEMT INQUIRE MONITOR 241
TSYSTEM option
INQUIRE WORKREQUEST command 353
TTI
CEMT INQUIRE NETNAME 247
CEMT INQUIRE TERMINAL 307
CEMT SET NETNAME 429
CEMT SET TERMINAL 454
CEOT 468, 469
TTISTATUS
CEMT INQUIRE NETNAME 247

TTISTATUS *(continued)*
CEMT INQUIRE TERMINAL 307
TTTT
CSPG 560
TWAIT
CEMT INQUIRE DB2CONN 174
CEMT INQUIRE DB2ENTRY 178
CEMT SET DB2CONN 389
CEMT SET DB2ENTRY 394
TX
CEMT INQUIRE DB2CONN 169
CEMT INQUIRE DB2ENTRY 176
CEMT SET DB2CONN 385
CEMT SET DB2ENTRY 392
TXID
CEMT INQUIRE DB2CONN 168
CEMT INQUIRE DB2ENTRY 176
CEMT SET DB2CONN 384
CEMT SET DB2ENTRY 391
TYPE
CEMT INQUIRE DOCTEMPLATE 188
CEMT INQUIRE ENQ 203
CEMT INQUIRE JMODEL 227
CEMT INQUIRE JOURNALNAME 229
CEMT INQUIRE REQUESTMODEL 264
CEMT INQUIRE TDQUEUE 300
CEMT INQUIRE UOWENQ 336
CEMT INQUIRE UOWLINK 340
TYPETERM
CEDA 101

U

U
CEMT INQUIRE TASK 286
UDATAKEY
CEMT INQUIRE TRANSACTION 312
UDSASIZE
CEMT INQUIRE DSAS 191
UEXECKEY
CEMT INQUIRE PROGRAM 260
CEMT PERFORM JVMPOOL command 363
UKOPEN
CEMT INQUIRE TASK 286
UNATTEMPTED
CEMT INQUIRE CONNECTION 158
UNAUTH
CEMT INQUIRE CORBASERVER 167
UNAVAILABLE
CEMT INQUIRE DSNAME 195
CEMT INQUIRE UOWLINK 339
CEMT INQUIRE VTAM 345
CEMT SET DSNAME 404
INQUIRE TCPIPSERVICE 293
UNBALANCD PARENS AT XXXXX
CMSG 533
UNCONNECTED
CEMT INQUIRE TSPOOL 319
CEMT INQUIRE UOWLINK 339
UNDEFINED
CEMT INQUIRE UOWDSNFAIL 330
UNDETERMINED
CEMT INQUIRE DSNAME 198
UNENABLED
CEMT INQUIRE FILE 212
UNKNOWN
CEMT INQUIRE UOWLINK 340
UNQUIESCED
CEMT INQUIRE DSNAME 197
CEMT SET DSNAME 404
UNREGISTERED
CEMT INQUIRE VTAM 345
INQUIRE TCPIPSERVICE 293
UNSUP OPRT TRMNL XXXXXXXX
CMSG 533
UNSUP TERMINAL XXXXXXXX
CMSG 533
UOW
CEMT INQUIRE DB2CONN 168
CEMT INQUIRE DB2ENTRY 175
CEMT INQUIRE ENQ 204
CEMT INQUIRE TASK 286
CEMT INQUIRE UOW 326
CEMT INQUIRE UOWDSNFAIL 333
CEMT INQUIRE UOWENQ 337
CEMT INQUIRE UOWLINK 340
CEMT SET DB2CONN 384
CEMT SET transaction 458
CEMT transaction 323
UOWACTION
CEMT INQUIRE CONNECTION 162
CEMT INQUIRE DSNAME 198
UOWDSNFAIL
CEMT INQUIRE transaction 328
CEMT INQUIRE UOWDSNFAIL 333
UOWENQ
CEMT transaction 333
UOWLINK
CEMT INQUIRE transaction 337
CEMT INQUIRE UOWLINK 340
CEMT SET transaction 459
UOWSTATE
CEMT INQUIRE UOW 326
UPDATE
CEMT INQUIRE FILE 215
CEMT SET FILE 410
UPDATEMODEL
CEMT INQUIRE FILE 215
UPDATESTATUS
CEMT INQUIRE FILE 215
uppercase input to transactions 4
URID
CEMT INQUIRE EXCI 208
CEMT INQUIRE UOWLINK 340
URIMAP
CEDA 101
CEMT DISCARD 138
CEMT INQUIRE transaction 340
CEMT INQUIRE WEBSERVICE 349
CEMT PERFORM STATISTICS 369
CEMT SET transaction 460

URM
CEMT SET TCPIPSERVICE 449
USAGE
CEMT INQUIRE URIMAP 342
USECOUNT
CEMT INQUIRE PROGRAM 262
CEMT INQUIRE STREAMNAME 270
USERAUTH option
CEMT INQUIRE IPCONN command 223
USERID 476
CEMT INQUIRE BRFACILITY 150
CEMT INQUIRE DB2CONN 169
CEMT INQUIRE DB2ENTRY 177
CEMT INQUIRE NETNAME 247
CEMT INQUIRE TASK 287
CEMT INQUIRE TDQUEUE 300
CEMT INQUIRE TERMINAL 307
CEMT INQUIRE UOW 327
CEMT INQUIRE URIMAP 343
CEMT SET DB2CONN 385
CEMT SET DB2ENTRY 392
USERTABLE
CEMT INQUIRE FILE 215
CEMT SET FILE 411

V

VALID
CEMT INQUIRE DSNAME 198
VALIDATIONST
CEMT INQUIRE WEBSERVICE 349, 464
VALIDITY
CEMT INQUIRE DSNAME 198
value
CEKL SET TASK 119
CEMT INQUIRE BEAN 146
CEMT INQUIRE BRFACILITY 148
CEMT INQUIRE CONNECTION 155
CEMT INQUIRE CORBASERVER 165
CEMT INQUIRE DJAR 184
CEMT INQUIRE DSNAME 191
CEMT INQUIRE FILE 208
CEMT INQUIRE HOST 217
CEMT INQUIRE IPCONN 219
CEMT INQUIRE MODENAME 237
CEMT INQUIRE NETNAME 242
CEMT INQUIRE PARTNER 247
CEMT INQUIRE PROFILE 253
CEMT INQUIRE PROGRAM 254
CEMT INQUIRE REQUESTMODEL 263
CEMT INQUIRE STREAMNAME 269
CEMT INQUIRE SYDUMPCODE 271
CEMT INQUIRE TASK 280
CEMT INQUIRE TCLASS 287
CEMT INQUIRE TDQUEUE 296
CEMT INQUIRE TERMINAL 301
CEMT INQUIRE TRANSACTION 307
CEMT INQUIRE TRDUMPCODE 313
CEMT INQUIRE TSMODEL 316
CEMT INQUIRE TSPOOL 318
CEMT INQUIRE TSQUEUE 320
value *(continued)*
CEMT INQUIRE UOW 324
CEMT INQUIRE UOWLINK 337
CEMT INQUIRE URIMAP 341
CEMT INQUIRE WORKREQUEST 351
CEMT SET BRFACILITY 374
CEMT SET CONNECTION 376
CEMT SET CORBASERVER 382
CEMT SET DOCTEMPLATE 399
CEMT SET DSNAME 401
CEMT SET ENQMODEL 405
CEMT SET FILE 407
CEMT SET HOST 412
CEMT SET IPCONN 414
CEMT SET JOURNALNAME 418
CEMT SET LINE 422
CEMT SET MODENAME 424
CEMT SET NETNAME 428
CEMT SET PIPELINE 430
CEMT SET PROGRAM 433
CEMT SET SYDUMPCODE 439
CEMT SET TASK 445
CEMT SET TCLASS 446
CEMT SET TERMINAL 452
CEMT SET TRANSACTION 455
CEMT SET TRDUMPCODE 456
CEMT SET TSQUEUE 458
CEMT SET UOW 459
CEMT SET UOWLINK 460
CEMT SET WEBSERVICE 464
CEMT SET WORKREQUEST 465
value option
CEMT INQUIRE REQUESTMODEL 264
values
hexadecimal 131
view Java debugging profile screen
application debugging profile manager
3270 interface 51
View Java profile page
application debugging profile manager
web interface 33
view LE debugging profile screen
application debugging profile manager
3270 interface 50
View LE profile page
application debugging profile manager
web interface 32
viewing debugging profiles
with the 3270 interface 41
with the web interface 24
VSAM
CEMT INQUIRE DSNAME 193
CEMT INQUIRE FILE 210
VTAM
CEMT INQUIRE CONNECTION 157
CEMT INQUIRE transaction 344
CEMT INQUIRE VTAM 346
CEMT PERFORM STATISTICS 369
CEMT SET transaction 461
VTAM ACB (access control block) 344, 461
VTAM terminals, acquiring 428, 452

W

WAIT
  CEMT INQUIRE TASK 285
  CEMT INQUIRE TDQUEUE 299
  CEMT INQUIRE TRANSACTION 310
WAITCAUSE
  CEMT INQUIRE UOW 327
WAITCOMMIT
  CEMT INQUIRE UOW 327
WAITER
  CEMT INQUIRE ENQ 203
  CEMT INQUIRE UOWENQ 335
WAITFORGET
  CEMT INQUIRE UOW 327
WAITING
  CEMT INQUIRE ENQMODEL 206
  CEMT INQUIRE UOW 328
WAITRMI
  CEMT INQUIRE UOW 327
WAITSTATE
  CEMT INQUIRE UOW 328
web interface
  application debugging profile manager 17
Web support 463
  CEMT INQUIRE transaction 346
  CEMT SET transaction 463
Web support inquiry 346
WEBSERVICE
  CEDA 101
  CEMT DISCARD 138
  CEMT INQUIRE URIMAP 343
  CEMT PERFORM STATISTICS 369
  CEMT SET WEBSERVICE 463
work requests
  CEMT INQUIRE requests 350
  CEMT SET requests 464
WORKREQUEST
  CEMT SET transaction 464
WORKREQUEST command
  CEMT INQUIRE transaction 350
WORKREQUEST option
  INQUIRE WORKREQUEST command 353
WORKTYPE option
  INQUIRE WORKREQUEST command 353
write to console operator (CWTO) 563
WSBIND
  CEMT INQUIRE WEBSERVICE 349
WSDIR
  CEMT INQUIRE PIPELINE 251
  CEMT INQUIRE WEBSERVICE 348
WSDLFILE
  CEMT INQUIRE WEBSERVICE 349

X

XCF
  CEMT INQUIRE CONNECTION 157
XCFGROUP
  CEMT INQUIRE IRC 225
XID
  CEMT INQUIRE UOWLINK 340
XLNSTATUS
  CEMT INQUIRE CONNECTION 162
XLT
  CEMT PERFORM SHUTDOWN 366
XM
  CEMT INQUIRE CONNECTION 157
XNOTDONE
  CEMT INQUIRE CONNECTION 162
XOK
  CEMT INQUIRE CONNECTION 162
XOPDIRECTST
  CEMT INQUIRE PIPELINE 251
  CEMT INQUIRE WEBSERVICE 349
XOPSUPPORTST
  CEMT INQUIRE PIPELINE 251
  CEMT INQUIRE WEBSERVICE 350
XRF and the CEBT transaction 79
XRF and the CEKL transaction 115

Y

YES
  CMSG 519, 520
YY.DDD
  CMSG 516
YYYY
  CSPG 557
YYYY.DDD
  CMSG 518

Z

z/OS UNIXFILE
  CEMT INQUIRE DJAR 185
ZERO LENGTH MSG AT ''
  CMSG 533

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.

© Copyright IBM Corp. 1982, 2011

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. 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 or registered trademarks of Oracle and/or its affiliates.

Other product and service names might be trademarks of IBM or other companies.

© Copyright IBM Corp. 1982, 2011

Readers' Comments — We'd Like to Hear from You

CICS Transaction Server for z/OS
CICS Supplied Transactions
Version 3 Release 2

Publication No. SC34-6817-03

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.
- Send a fax to the following number: +44–1962–816151
- Send your comments via email to: idrcf@hursley.ibm.com

If you would like a response from IBM, please fill in the following information:

Name	Address
Company or Organization	
Phone No.	Email address

Readers' Comments — We'd Like to Hear from You
SC34-6817-03

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM United Kingdom Limited
User Technologies Department (MP095)
Hursley Park
Winchester
Hampshire
SO21 2JN
United Kingdom

Fold and Tape | **Please do not staple** | Fold and Tape

SC34-6817-03

Cut or Fold
Along Line

Product Number: 5655-M15

SC34-6817-03

Spine information:

CICS Transaction Server for z/OS CICS Supplied Transactions Version 3 Release 2