

CICS Transaction Server for z/OS



# CICS Supplied Transactions

*Version 3 Release 1*



CICS Transaction Server for z/OS



# CICS Supplied Transactions

*Version 3 Release 1*

**Note!**

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 669.

This edition applies to Version 3 Release 1 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. Make sure you are using the correct edition for the level of the product.

© Copyright IBM Corporation 1982, 2011.

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

---

# Contents

<b>Preface</b> . . . . .	xii
What this book is about . . . . .	xii
Who should read this book . . . . .	xii
What you need to know to understand this book . . . . .	xii
How to use this book . . . . .	xii
<b>Summary of changes</b> . . . . .	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 . . . . .	xiii
Changes for CICS Transaction Server for z/OS, Version 2 Release 2 . . . . .	xiii
Changes for CICS Transaction Server for z/OS, Version 2 Release 1 . . . . .	xiv
Changes for the CICS Transaction Server for OS/390 Version 1 Release 3 edition . . . . .	xv
Changes for the CICS Transaction Server for OS/390 Version 1 Release 2 edition . . . . .	xvi
Changes for the CICS Transaction Server for OS/390 Version 1 Release 1 edition . . . . .	xvi
<b>Chapter 1. Using CICS supplied transactions</b> . . . . .	1
How to start or stop a transaction . . . . .	2
Syntax notation and conventions used . . . . .	2
Minimum abbreviation of keywords . . . . .	3
Uppercase input to transactions . . . . .	4
Terminal operation . . . . .	4
\$ (the dollar symbol) . . . . .	4
Operator security . . . . .	5
Terminal operator . . . . .	5
Supervisory terminal operator . . . . .	5
Master terminal operator . . . . .	6
MRO and ISC support . . . . .	7
BMS partitions . . . . .	7
Using the system console . . . . .	9
Rules for console entry . . . . .	10
MODIFY command . . . . .	10
REPLY command . . . . .	11
Example of a conversation using CONVERSE . . . . .	11
Example of a conversation using WRITE/READ . . . . .	11
Using TSO consoles . . . . .	12
Using the KILL option for purging transactions . . . . .	12
Using CEMT to terminate a task . . . . .	13
Using the console to terminate a task . . . . .	13
Consequences of using KILL . . . . .	14
<b>Chapter 2. The application debugging profile manager</b> . . . . .	15
Using the application debugging profile manager Web interface . . . . .	16
Configuring access to the application debugging profile manager Web interface . . . . .	16
Using the application debugging profile manager 3270 interface . . . . .	37
<b>Chapter 3. CBAM—BTS browser</b> . . . . .	57
<b>Chapter 4. CCRL — certificate revocation lists transaction</b> . . . . .	59

<b>Chapter 5. CDBC—database control menu.</b>	61
Connecting CICS to DBCTL	62
Disconnecting CICS from DBCTL	62
Specifying a DBCTL system identifier (DBCTLID)	63
Using the HELP panel	64
<b>Chapter 6. CDBI—database control inquiry</b>	65
Using the CDBI HELP panel	66
<b>Chapter 7. CDBM—database control interface</b>	67
Using CDBM Syntax	67
Using DFHDBFK - The CDBM GROUP command file	70
Record layout in the CDBM GROUP command file.	70
The MAINTENANCE panel for DFHDBFK	70
Using the CDBM HELP panels	72
<b>Chapter 8. CEBR—temporary storage browse</b>	75
Using the CEBR HELP panel	76
<b>Chapter 9. CEBT—master terminal (alternate CICS)</b>	79
Minimum abbreviation of keywords	79
CEBT INQUIRE and SET commands.	79
CEBT INQUIRE AUXTRACE	80
CEBT INQUIRE DUMP	82
CEBT INQUIRE GTFTRACE	83
CEBT INQUIRE INTTRACE	84
CEBT INQUIRE SURVEILLANCE	85
CEBT INQUIRE TAKEOVER	86
CEBT PERFORM commands	87
CEBT PERFORM SHUTDOWN.	88
CEBT PERFORM SNAP	89
CEBT PERFORM TAKEOVER	90
CEBT SET AUXTRACE.	91
CEBT SET DUMP.	93
CEBT SET GTFTRACE.	94
CEBT SET INTTRACE	95
CEBT SET SURVEILLANCE	96
CEBT SET TAKEOVER.	97
<b>Chapter 10. CECI—command-level interpreter</b>	99
Using the CECI HELP panel	100
Command-level syntax-checking transaction (CECS)	101
<b>Chapter 11. CEDA—resource definition online</b>	103
Command syntax	105
CSD compatibility	112
CSD compatibility panel	112
Security checking	114
CEDA HELP panels.	114
CEDB and CEDC	119
CEDB.	119
CEDC.	120
<b>Chapter 12. CEDF &amp; CEDX — the execution diagnostic facility</b>	121
Using the CEDF transaction	121
Command syntax	121

Command options . . . . .	121
Beginning an EDF session (single-screen mode) . . . . .	122
Beginning an EDF session (dual-screen mode) . . . . .	123
Monitoring remote transactions . . . . .	123
Invoking CEBR and CECI from CEDF . . . . .	123
Using the CEDX transaction . . . . .	124
Command syntax . . . . .	125
Command options . . . . .	125
<b>Chapter 13. CEKL—master terminal (emergency use) . . . . .</b>	<b>127</b>
Minimum abbreviation of keywords . . . . .	127
CEKL INQUIRE TASK . . . . .	128
CEKL SET TASK . . . . .	132
<b>Chapter 14. CEMT—master terminal . . . . .</b>	<b>135</b>
Starting the CEMT transaction . . . . .	135
System parameters . . . . .	138
Program function (PF) keys . . . . .	138
Minimum abbreviation of keywords . . . . .	139
ALL option . . . . .	139
CLASS option . . . . .	139
Family of resources (using * and + symbols) . . . . .	139
Subdefining resource groups . . . . .	140
Overtyping a display . . . . .	140
Tab key . . . . .	141
Scrolling symbol (+ sign) . . . . .	141
Blank fields in a display . . . . .	141
Using the ? key . . . . .	141
Expanded panels . . . . .	142
Hexadecimal values . . . . .	143
Variables . . . . .	143
Invoking CEMT from an application program . . . . .	144
Security checking . . . . .	144
CEMT DISCARD commands . . . . .	145
CEMT DISCARD . . . . .	147
CEMT INQUIRE commands . . . . .	152
CEMT INQUIRE AUTINSTMODEL . . . . .	154
CEMT INQUIRE AUTOINSTALL . . . . .	156
CEMT INQUIRE AUXTRACE . . . . .	159
CEMT INQUIRE BEAN . . . . .	161
CEMT INQUIRE BRFACILITY . . . . .	163
CEMT INQUIRE CFDTPOOL . . . . .	167
CEMT INQUIRE CLASSCACHE . . . . .	169
CEMT INQUIRE CONNECTION . . . . .	173
CEMT INQUIRE CORBASERVER . . . . .	182
CEMT INQUIRE DB2CONN . . . . .	186
CEMT INQUIRE DB2ENTRY . . . . .	193
CEMT INQUIRE DB2TRAN . . . . .	198
CEMT INQUIRE DELETSHIPED . . . . .	200
CEMT INQUIRE DISPATCHER . . . . .	202
CEMT INQUIRE DJAR . . . . .	205
CEMT INQUIRE DOCTEMPLATE . . . . .	208
CEMT INQUIRE DSAS . . . . .	211
CEMT INQUIRE DSNAME . . . . .	214
CEMT INQUIRE DUMPDS . . . . .	223
CEMT INQUIRE ENQ . . . . .	225

	CEMT INQUIRE ENQMODEL . . . . .	230
	CEMT INQUIRE EXCI . . . . .	232
	CEMT INQUIRE FEPI . . . . .	234
	CEMT INQUIRE FILE . . . . .	235
	CEMT INQUIRE GTFTRACE . . . . .	242
	CEMT INQUIRE HOST . . . . .	244
	CEMT INQUIRE INTTRACE . . . . .	246
	CEMT INQUIRE IRBATCH . . . . .	248
	CEMT INQUIRE IRC . . . . .	249
	CEMT INQUIRE JMODEL . . . . .	251
	CEMT INQUIRE JOURNALNAME . . . . .	253
	CEMT INQUIRE JVM . . . . .	256
	CEMT INQUIRE JVMPOOL . . . . .	259
	CEMT INQUIRE LINE . . . . .	261
	CEMT INQUIRE MODENAME . . . . .	263
	CEMT INQUIRE MONITOR . . . . .	265
	CEMT INQUIRE NETNAME . . . . .	269
	CEMT INQUIRE PARTNER . . . . .	275
	CEMT INQUIRE PIPELINE . . . . .	277
	CEMT INQUIRE PROCESSTYPE . . . . .	280
	CEMT INQUIRE PROFILE . . . . .	282
	CEMT INQUIRE PROGRAM . . . . .	283
	CEMT INQUIRE REQUESTMODEL . . . . .	291
	CEMT INQUIRE RRMS . . . . .	294
	CEMT INQUIRE STATISTICS . . . . .	295
	CEMT INQUIRE STREAMNAME . . . . .	299
	CEMT INQUIRE SYDUMPCODE . . . . .	301
	CEMT INQUIRE SYSTEM . . . . .	305
	CEMT INQUIRE TASK . . . . .	311
	CEMT INQUIRE TCLASS . . . . .	317
	CEMT INQUIRE TCPIP . . . . .	319
	CEMT INQUIRE TCPIPSERVICE . . . . .	320
	CEMT INQUIRE TDQUEUE . . . . .	326
	CEMT INQUIRE TERMINAL . . . . .	331
	CEMT INQUIRE TRANSACTION . . . . .	338
	CEMT INQUIRE TRDUMPCODE . . . . .	344
	CEMT INQUIRE TSMODEL . . . . .	347
	CEMT INQUIRE TSPool . . . . .	350
	CEMT INQUIRE TSQUEUE / TSQNAME . . . . .	352
	CEMT INQUIRE UOW . . . . .	356
	CEMT INQUIRE UOWDSNFAIL . . . . .	361
	CEMT INQUIRE UOWENQ . . . . .	366
	CEMT INQUIRE UOWLINK . . . . .	371
	CEMT INQUIRE URIMAP . . . . .	375
	CEMT INQUIRE VTAM . . . . .	380
	CEMT INQUIRE WEB . . . . .	382
	CEMT INQUIRE WEBSERVICE . . . . .	383
	CEMT INQUIRE WORKREQUEST . . . . .	387
	CEMT PERFORM commands . . . . .	391
	CEMT PERFORM CLASSCACHE . . . . .	392
	CEMT PERFORM CORBASERVER . . . . .	396
	CEMT PERFORM DELETSHIPED . . . . .	398
	CEMT PERFORM DJAR . . . . .	399
	CEMT PERFORM DUMPISNAP . . . . .	400
	CEMT PERFORM ENDAFFINITY . . . . .	401
	CEMT PERFORM PIPELINE . . . . .	402

CEMT PERFORM RESET . . . . .	403
CEMT PERFORM SECURITY . . . . .	404
CEMT PERFORM SHUTDOWN . . . . .	405
CEMT PERFORM STATISTICS . . . . .	407
CEMT SET commands . . . . .	411
CEMT SET AUTOINSTALL . . . . .	412
CEMT SET AUXTRACE . . . . .	414
CEMT SET BRFACILITY . . . . .	416
CEMT SET CLASSCACHE . . . . .	417
CEMT SET CONNECTION . . . . .	418
CEMT SET CORBASERVER . . . . .	424
CEMT SET DB2CONN . . . . .	426
CEMT SET DB2ENTRY . . . . .	434
CEMT SET DB2TRAN . . . . .	439
CEMT SET DELETSHIPED . . . . .	440
CEMT SET DISPATCHER . . . . .	442
CEMT SET DSAS . . . . .	445
CEMT SET DSNAME . . . . .	446
CEMT SET DUMPDS . . . . .	451
CEMT SET ENQMODEL . . . . .	452
CEMT SET FEPI . . . . .	453
CEMT SET FILE . . . . .	454
CEMT SET GTFTRACE . . . . .	459
CEMT SET HOST . . . . .	460
CEMT SET INTTRACE . . . . .	461
CEMT SET IRC . . . . .	462
CEMT SET JOURNALNAME . . . . .	463
CEMT SET JVMPOOL . . . . .	464
CEMT SET LINE . . . . .	466
CEMT SET MODENAME . . . . .	468
CEMT SET MONITOR . . . . .	470
CEMT SET NETNAME . . . . .	473
CEMT SET PIPELINE . . . . .	476
CEMT SET PROCESSTYPE . . . . .	477
CEMT SET PROGRAM . . . . .	479
CEMT SET QUEUE . . . . .	482
CEMT SET STATISTICS . . . . .	483
CEMT SET SYDUMPCODE . . . . .	486
CEMT SET SYSTEM . . . . .	489
CEMT SET TASK . . . . .	493
CEMT SET TCLASS . . . . .	495
CEMT SET TCPIP . . . . .	496
CEMT SET TCPIPSERVICE . . . . .	498
CEMT SET TDQUEUE . . . . .	500
CEMT SET TERMINAL . . . . .	502
CEMT SET TRANSACTION . . . . .	506
CEMT SET TRDUMPCODE . . . . .	508
CEMT SET TSQUEUE . . . . .	510
CEMT SET UOW . . . . .	511
CEMT SET UOWLINK . . . . .	512
CEMT SET URIMAP . . . . .	513
CEMT SET VTAM . . . . .	515
CEMT SET WEB . . . . .	517
CEMT SET WEBSERVICE . . . . .	518
CEMT SET WORKREQUEST . . . . .	519

<b>Chapter 15. CEOT—terminal status</b> . . . . .	521
Using CEOT in inquiry mode . . . . .	521
CEOT in set mode . . . . .	522
<b>Chapter 16. CESF—sign off</b> . . . . .	525
<b>Chapter 17. CESN—sign on</b> . . . . .	527
IBM 3270 family display devices . . . . .	528
Non-3270 family terminal. . . . .	529
Command options . . . . .	529
Signing off with CESN. . . . .	530
<b>Chapter 18. CEST—supervisory terminal</b> . . . . .	531
<b>Chapter 19. CETR—trace control</b> . . . . .	533
Using component trace options . . . . .	534
Using Transaction and terminal-related trace options . . . . .	537
Using JVM trace options . . . . .	538
Using CETR HELP panels . . . . .	541
HELP panels for components . . . . .	543
HELP panels for terminal and transaction trace . . . . .	546
HELP panels for JVM trace options . . . . .	548
<b>Chapter 20. CIDP — inactivate debugging profiles</b> . . . . .	551
<b>Chapter 21. CIND—in-doubt testing tool</b> . . . . .	553
Running the CIND in-doubt tool . . . . .	554
Inquiring on the status of CIND . . . . .	555
Running the in-doubt tool using an EXEC CICS LINK command . . . . .	555
Setting a single transaction in-doubt . . . . .	555
CIND Messages . . . . .	555
<b>Chapter 22. CLER—Language Environment run-time options</b> . . . . .	557
<b>Chapter 23. CMAC—messages and codes display</b> . . . . .	559
<b>Chapter 24. CMSG—message switching</b> . . . . .	563
Sending a message . . . . .	563
Canceling a message . . . . .	564
Message-switching continuation . . . . .	565
Message delivery . . . . .	566
Subsystems and terminal systems . . . . .	566
IBM 3270 display devices . . . . .	567
IBM 3767 Communication Terminal . . . . .	567
Sequential terminal . . . . .	567
CMSG command . . . . .	568
CMSG command options. . . . .	568
Message-switching responses . . . . .	579
CMSG examples. . . . .	586
<b>Chapter 25. CPIA—unit of work resynchronization</b> . . . . .	589
<b>Chapter 26. CREA—create REQUESTMODELS for enterprise beans</b> . . . . .	591
Function: . . . . .	591
Using CREA . . . . .	592
Starting CREA . . . . .	592

#

TranID Specification . . . . .	593
REQUESTMODEL Specification . . . . .	597
Results . . . . .	600
CREC . . . . .	602
Using CREC . . . . .	602
<b>Chapter 27. CRTE—remote transactions . . . . .</b>	<b>603</b>
<b>Chapter 28. CSFE—terminal and system test . . . . .</b>	<b>605</b>
Using CSFE terminal test . . . . .	605
Tracing with CSFE . . . . .	606
Terminal builder trace . . . . .	606
Using CSFE debug functions . . . . .	606
Storage freeze . . . . .	606
Storage violation trap . . . . .	607
Global trap/trace program exit and global kernel exit . . . . .	608
<b>Chapter 29. CSPG—page retrieval . . . . .</b>	<b>609</b>
Retrieve a page . . . . .	609
Single keystroke retrieval (SKR) . . . . .	611
Page retrieval for logical devices . . . . .	611
Page retrieval and partitions . . . . .	612
Display information about logical messages . . . . .	612
Message chaining with CSPG . . . . .	612
CSPG message termination. . . . .	615
Copying pages with CSPG . . . . .	616
Using Autopage with CSPG . . . . .	616
Displaying routed pages with CSPG . . . . .	617
<b>Chapter 30. CWTO—write to console operator . . . . .</b>	<b>619</b>
<b>Chapter 31. DSNC—CICS DB2 transaction DSNC . . . . .</b>	<b>621</b>
<b>Appendix. List of CICS transactions . . . . .</b>	<b>623</b>
<b>Bibliography . . . . .</b>	<b>629</b>
The CICS Transaction Server for z/OS library . . . . .	629
The entitlement set . . . . .	629
PDF-only books . . . . .	629
Other CICS books . . . . .	631
Books from related libraries. . . . .	631
Determining if a publication is current . . . . .	632
<b>Accessibility . . . . .</b>	<b>633</b>
<b>Index . . . . .</b>	<b>635</b>
<b>Notices . . . . .</b>	<b>669</b>
Trademarks. . . . .	670
<b>Sending your comments to IBM . . . . .</b>	<b>671</b>



---

# 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	<i>CICS® Business Transaction Services</i>
CDBC	<i>CICS IMS™ Database Control Guide</i>
CDBI	<i>CICS Data Areas</i>
CEBR	<i>CICS Application Programming Guide</i>
CECI	<i>CICS Application Programming Guide</i>
CECS	<i>CICS Application Programming Guide</i>
CEDA	<i>CICS Resource Definition Guide</i>
CEDF	<i>CICS Application Programming Guide</i>
CETR	<i>CICS Problem Determination Guide</i>
CRTE	<i>CICS RACF® Security Guide</i>
CSFE	<i>CICS Problem Determination Guide</i>
DSNC	<i>CICS DB2® Guide</i>

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

### **Notes on terminology**

- **CICS** refers to IBM® CICS Transaction Server for z/OS®, Version 3 Release 1
- **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, SC34-5724-00. Changes from that edition are marked by vertical bars in the left margin.

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

---

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

The more significant changes for this edition are:

- 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.

---

### 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 15 has been added.
  - Chapter 20, “CIDP — inactivate debugging profiles,” on page 551 has been added.
- New commands :
  - “CEMT INQUIRE WORKREQUEST” on page 387 added to CEMT
  - “CEMT SET WORKREQUEST” on page 519 added to CEMT
  - “CEMT INQUIRE CLASSCACHE” on page 169 added to CEMT
  - “CEMT SET CLASSCACHE” on page 417 added to CEMT
  - “CEMT PERFORM CLASSCACHE” on page 392 added to CEMT
  - “CEMT INQUIRE JVM” on page 256 added to CEMT
- New options:
  - “DEBUGTOOL” on page 306 added to CEMT INQUIRE SYSTEM
  - “DEBUG” on page 490 added to CEMT SET SYSTEM
  - “NODEBUG” on page 491 added to CEMT SET SYSTEM

---

### 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 186, the definitions of parameters DB2ID, PRIORITY, TCBLIMIT and TCBS have changed.
- In “CEMT INQUIRE DB2ENTRY” on page 193, the definition of the PRIORITY parameter has changed.
- In “CEMT SET DB2CONN” on page 426, the definitions of parameters DB2ID, PRIORITY, TCBLIMIT and TCBS have changed.
- In “CEMT SET DB2ENTRY” on page 434, the definition of the PRIORITY parameter has changed.
- In “CEMT INQUIRE TCPIPSERVICE” on page 320, there are changes to AUTHENTICATE and the description of its CVDA values.
- There are new transactions CREA and CREC, see Chapter 26, “CREA—create REQUESTMODELS for enterprise beans,” on page 591 for details.
- New options:
  - “DB2GROUPLD” on page 189 added to CEMT INQUIRE DB2CONN
  - “RESYNCMEMBER” on page 191 added to CEMT INQUIRE DB2CONN
  - “PLAN” on page 198 added to CEMT INQUIRE DB2TRAN
  - “PLANEXITNAME” on page 199 added to CEMT INQUIRE DB2TRAN
  - “SCAN” on page 397 added to CEMT PERFORM CORBASERVER
  - “DB2GROUPLD” on page 430 added to CEMT SET DB2CONN
  - “RESYNCMEMBER” on page 431 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 161
  - “CEMT INQUIRE CORBASERVER” on page 182
  - “CEMT INQUIRE DJAR” on page 205
  - “CEMT INQUIRE JVMPOOL” on page 259
  - “CEMT PERFORM CORBASERVER” on page 396
  - “CEMT INQUIRE UOWENQ” on page 366
  - “CEMT DISCARD commands” on page 145
  - “CEMT DISCARD commands” on page 145
  - “CEMT SET CORBASERVER” on page 424
  - “CEMT SET JVMPOOL” on page 464
- New chapters:
  - Chapter 22, “CLER—Language Environment run-time options,” on page 557
  - Chapter 27, “CRTE—remote transactions,” on page 603
- 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” on page 289 added to CEMT INQUIRE PROGRAM
  - “BEANNAME” on page 292 added to CEMT INQUIRE REQUESTMODEL
  - “CORBASERVER” on page 292 added to CEMT INQUIRE REQUESTMODEL
  - “INTERFACE” on page 292 added to CEMT INQUIRE REQUESTMODEL
  - “INTFACETYPE” on page 292 added to CEMT INQUIRE REQUESTMODEL
  - “MODULE” on page 292 added to CEMT INQUIRE REQUESTMODEL
  - “OPERATION” on page 293 added to CEMT INQUIRE REQUESTMODEL
  - “TYPE” on page 293 added to CEMT INQUIRE REQUESTMODEL
  - “ACTSOCKETS” on page 319 added to CEMT INQUIRE TCPIP

- “MAXSOCKETS” on page 319 added to CEMT INQUIRE TCPIP
- “AUTHENTICATE” on page 321 added to CEMT INQUIRE TCPIPSERVICE
- “DNSGROUP” on page 322 added to CEMT INQUIRE TCPIPSERVICE
- “DNSSTATUS” on page 322 added to CEMT INQUIRE TCPIPSERVICE
- “GRPCRITICAL” on page 323 added to CEMT INQUIRE TCPIPSERVICE
- “OTSTIMEOUT” on page 341 added to CEMT INQUIRE TRANSACTION
- “OTSTID” on page 358 added to CEMT INQUIRE UOW
- “HOST” on page 372 added to CEMT INQUIRE UOWLINK
- “CORBASERVER” on page 409 added to CEMT PERFORM STATISTICS
- “JVMPOOL” on page 409 added to CEMT PERFORM STATISTICS
- “REQUESTMODEL” on page 409 added to CEMT PERFORM STATISTICS
- “TCPIP” on page 410 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” on page 497 added to CEMT SET TCPIP
- “DNSSTATUS” on page 498 added to CEMT SET TCPIPSERVICE
- “NOUCTRAN” on page 523 added to CEOT IN SET MODE
- “TRANIDONLY” on page 523 added to CEOT IN SET MODE
- “UCTRAN” on page 523 added to CEOT IN SET MODE
- New values
  - “OTSCoord” on page 360 added to WAITCAUSE in CEMT INQUIRE UOW

---

## Changes for the CICS Transaction Server for OS/390 Version 1 Release 3 edition

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

- The CEMT transaction is extended to support the new resource types:
  - DOCTEMPLATE
  - ENQMODEL
  - PROCESSTYPE
  - REQUESTMODEL
  - TCPIPSERVICE
  - TSMODEL

also added to CEMT are the following:

- INQUIRE CFDTPOOL
- INQUIRE ENQ
- INQUIRE RRMS
- INQUIRE TCPIP and SET TCPIP
- INQUIRE TSPool
- INQUIRE TSQNAME and SET TSQNAME
- INQUIRE TSQUEUE and SET TSQUEUE
- INQUIRE WEB and SET WEB

and there are changes to

- CEMT INQUIRE FILE and SET FILE in support of coupling facility data tables
- CEMT INQUIRE PROGRAM and CEMT SET PROGRAM to support running JAVA applications under the control of a Java Virtual Machine (JVM).
- The new CBAM transaction (described in “CEMT SET MODENAME” on page 468).
- The CDBM transaction is extended to support storing of commands for reuse, and issuing of stored commands.

---

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

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

Changes included:

- The CDEX transaction
- Addition of BRIDGE and IDENTIFIER to CEMT INQUIRE TASK.
- Addition of BREXIT and FACILITYLIKE to CEMT INQUIRE TRANSACTION.
- The CICS DB2 attachment facility is enhanced to provide resource definition online (RDO) support for DB2 resources as an alternative to resource control table (RCT) definitions. The following CEMT INQUIRE, and SET, commands are added:
  - CEMT INQUIRE|SET DB2CONN
  - CEMT INQUIRE|SET DB2ENTRY
  - CEMT INQUIRE|SET DB2TRAN
- The CEMT DISCARD command is extended to allow the resource definitions DB2CONN, DB2ENTRY, and DB2TRAN to be discarded.
- The CEDA check function is enhanced to perform consistency checks on the new DB2 objects.

---

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

This edition is based on the CICS/ESA 4.1 *CICS-Supplied Transactions* manual (SC33-0710-00).

For CICS Transaction Server for z/OS, the following changes have been made:

- Removal of local DL/I and CEMT INQUIRE and SET PITRACE
- Addition of CEMT DISCARD JMODEL, CEMT DISCARD JOURNALNAME, CEMT DISCARD TDQUEUE, CEMT INQUIRE EXCI (replacing CEMT INQUIRE IRBATCH), CEMT INQUIRE JMODEL, CEMT INQUIRE JOURNALNAME, CEMT INQUIRE STREAMNAME, CEMT INQUIRE UOW, CEMT INQUIRE UOWDSNFAIL, CEMT INQUIRE UOWENQ, CEMT INQUIRE UOWLINK, CEMT PERFORM ENDAFFINITY, CEMT SET JOURNALNAME, CEMT SET UOW, and CEMT SET UOWLINK commands
- New options on the CEMT INQUIRE CONNECTION, CEMT INQUIRE DSNAME, CEMT INQUIRE FILE, CEMT INQUIRE NETNAME, CEMT INQUIRE TASK, CEMT INQUIRE TDQUEUE, CEMT INQUIRE TERMINAL, CEMT PERFORM SHUTDOWN, CEMT PERFORM STATISTICS RECORD, CEMT SET CONNECTION, CEMT SET DSNAME, and CEMT SET TDQUEUE commands.

---

## 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. This book deals with those transactions that are supplied by CICS and which, in addition, 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 this book. For a more complete list of CICS transactions, including those that do not have an operator interface, see the “List of CICS transactions,” on page 623.

Function	Name	Page
CICS business transaction services (BTS) browser	CBAM	57
Command-level interpreter	CECI, CECS	99
Create and update certificate revocation lists (CRLs)	CCRL	59
Create REQUESTMODELS for enterprise beans	CREA, CREC	591
Database control inquiry	CDBI	65
Database control interface	CDBM	67
Database control menu	CDBC	61
Emergency Use	CEKL	127
Execution diagnostic facility	CEDF	121
	CEDX	121
In-doubt testing tool	CIND	553
Master terminal	CEMT	135
Master terminal (alternate CICS)	CEBT	79
Messages and codes display	CMAC	559
Message switching	CMSG	563
Page retrieval	CSPG	609
Remote transactions	CRTE	603
Resource definition online (RDO)	CEDA, CEDB, CEDC	103
Sign off	CESF	525
Sign on	CESN	527
Supervisory terminal	CEST	531
Temporary-storage browse	CEBR	75
Terminal status	CEOT	521
Trace control	CETR	533
Unit of work resynchronization	CPIA	589
Terminal and system test	CSFE	605
Write to console operator	CWTO	619

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

## introduction

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 4
- “Using the system console” on page 9
- “Using TSO consoles” on page 12

---

## 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
```

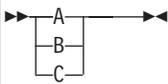
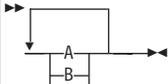
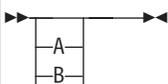
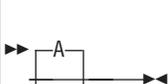
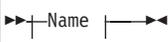
After a transaction has completed, press the CLEAR key to clear the screen in readiness for the next transaction. You can cancel any request by typing CANCEL on the command line.

---

## 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 <b>must</b> code.
	A set of alternatives—one of which you <b>must</b> code. You <b>may</b> code more than one of them, in any sequence.
	A set of alternatives—one of which you <b>may</b> code.
	A set of alternatives — any number (including none) of which you may code once, in any sequence.
	Alternatives where <b>A</b> is the default.
 <b>Name:</b> 	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.

## introduction

In the syntax displays on your screen (unless your terminal is uppercase only), and in most cases in this book, 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 book, the dollar symbol (\$) is used as a national currency symbol and is assumed to be assigned the EBCDIC code point X'5B'. In some countries a different currency symbol, for example the pound symbol (£), or

the yen symbol (¥), is assigned the same EBCDIC code point. In these countries, the appropriate currency symbol should be used instead of the dollar symbol.

## 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 *Resource Access Control Facility (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 may 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.

## introduction

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 9.

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 140 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 examples in the remainder of the book.

Unless otherwise stated, the information about the master terminal and its transactions given in this book 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 CICS system from the console, type:

```
MODIFY CICS,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 jjjjjjj 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.

**jjjjjjj** 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, CICSplex SM or CICS PM 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 14 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, CICSplex SM or CICS Performance Monitor. 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 127.

### 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 127 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.

## introduction

**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” on page 16.
- 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 20, “CIDP — inactivate debugging profiles,” on page 551.

---

## 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 all 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 all profiles” page.

If the sample analyzer program DFHWBADX, or an analyzer program with similar function, is used to handle requests for the application debugging profile manager, the path component of the URL gives 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 URIMAP definitions are used 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 all profiles page

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

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

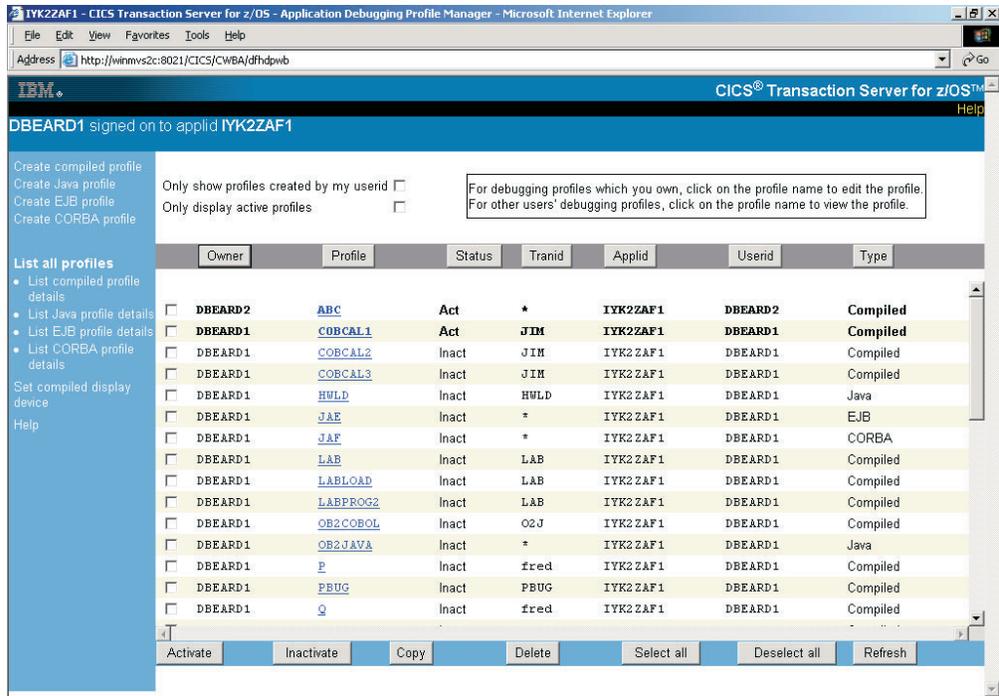


Figure 1. The List all profiles page

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 all profiles” page:

#### List compiled profile details

Lists only the profiles for Language Environment® (compiled language) programs

#### List Java™ profile details

Lists only the profiles for Java programs

#### List EJB profile details

Lists only the profiles for enterprise beans

#### List CORBA profile details

Lists only the profiles for stateless CORBA objects

The behavior of these pages is identical to the “List all 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 all profiles” and “List compiled profile details” pages only, displays the contents of the **program** field.

**Compile Unit** On the “List all profiles” and “List compiled profile details” 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 all profiles” and “List compiled profile details” pages only, displays the contents of the **Terminal** field.

**Type** On the “List all profiles” page only, displays the type of program specified in the debugging profile:

**CORBA**

CORBA object

**EJB** Enterprise bean

**Java** Java program

**Compiled**

Compiled language (Language Environment) program

**Netname** On the “List compiled profile details” page only, displays the contents of the **Netname** field.

**Class** On the “List Java profile details” and “List CORBA profile details” 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 profile details” 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 profile details” and “List CORBA profile details” 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 all 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 all 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 all profiles” page:

### Activate

Activate selected profiles. See “Activating debugging profiles with the Web interface” on page 21 for more information.

### Inactivate

Inactivate selected profiles. See “Inactivating debugging profiles with the Web interface” on page 22 for more information.

**Copy** Copy selected profiles. See “Copying debugging profiles with the Web interface” on page 23 for more information.

### Delete

Delete selected profiles. See “Deleting debugging profiles with the Web interface” on page 23 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 all 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 24 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 all 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 22. Before you save the profile, give it a new name.

#### If another user owns the profile

1. Copy the profile, following the steps described in “Copying debugging profiles with the Web interface” on page 23
2. Make any changes to the profile by following the steps described in “Changing a debugging profile with the Web interface” on page 22

---

## 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 program, 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 33 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 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 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 all 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.

**Notes:**

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 all 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 all 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 22. 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 all 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 all 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.

The screenshot shows a web browser window titled "IYK2ZAF1 - CICS Transaction Server for z/OS - Application Debugging Profile Manager - Microsoft Internet Explorer". The address bar shows a URL starting with "http://winmvs2c:8021/CICS/CWBA/dfvdpmib?page=dpwesort=dpwlfxxx&ftruser=xx&fracti=xx". The page content includes a blue header with "IBM" and "CICS® Transaction Server for z/OS™". Below the header, it says "DBEARD1 signed on to applid IYK2ZAF1". The main content area is titled "Create compiled profile" and contains several sections:
 

- Debugging profile**: A text input field followed by "owned by DBEARD1".
- CICS resources to debug**: A note "(use \* to specify generic values, e.g. \*, A\*, AB\*, etc)". It includes fields for Transaction, Applid (IYK2ZAF1), Program, Userid (DBEARD1), Compile unit, Termid, and Netname.
- Debug Tool Language Environment Options**: Includes a Test level dropdown menu (set to ALL), and text input fields for Command file, Prompt level (set to PROMPT), and Preference file.
- Other Language Environment Options**: A single text input field.

 A sidebar on the left lists options: "Create compiled profile" (with sub-links for Java, EJB, and CORBA), "List all profiles" (with sub-links for compiled, Java, EJB, and CORBA details), and "Help". At the bottom of the form, there are three buttons: "Create", "Replace", and "Save options as default".

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 debugging profile for a compiled language (Language Environment) program.

**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 dataset 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 dataset or a member of a partitioned data set.

**Language Environment options**

Specifies Language Environment run-time options for programs that match this profile. When a program is selected for debugging because it matches the profile, the run-time options specified will override other run-time options that you may have in effect. For more information about defining run-time 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 debugging profile for a compiled language (Language Environment) program.

**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.

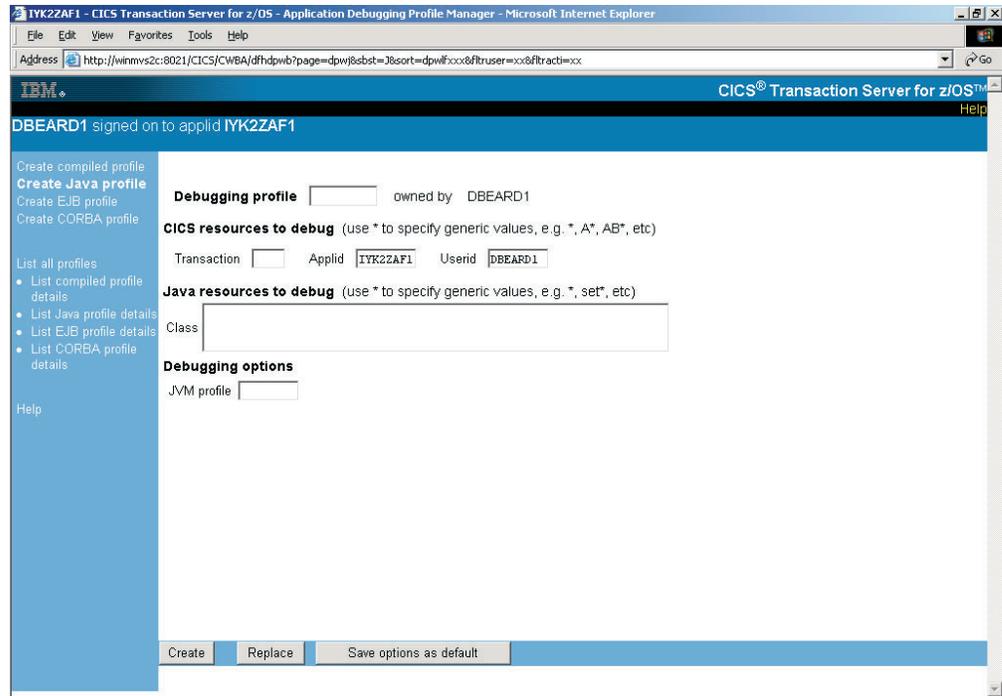


Figure 3. The Create Java profile page

## 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.

#### **AppId**

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:

<b>Java</b>	Select this value when you want to debug a Java program.
<b>EJB</b>	Select this value when you want to debug an enterprise bean.
<b>CORBA</b>	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 the *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 compiled profile page

Use the “View compiled profile” page to view the contents of a debugging profile for compiled language (Language Environment) programs. You cannot change the profile with this page. The information displayed is described in “The Create compiled profile page” on page 24.

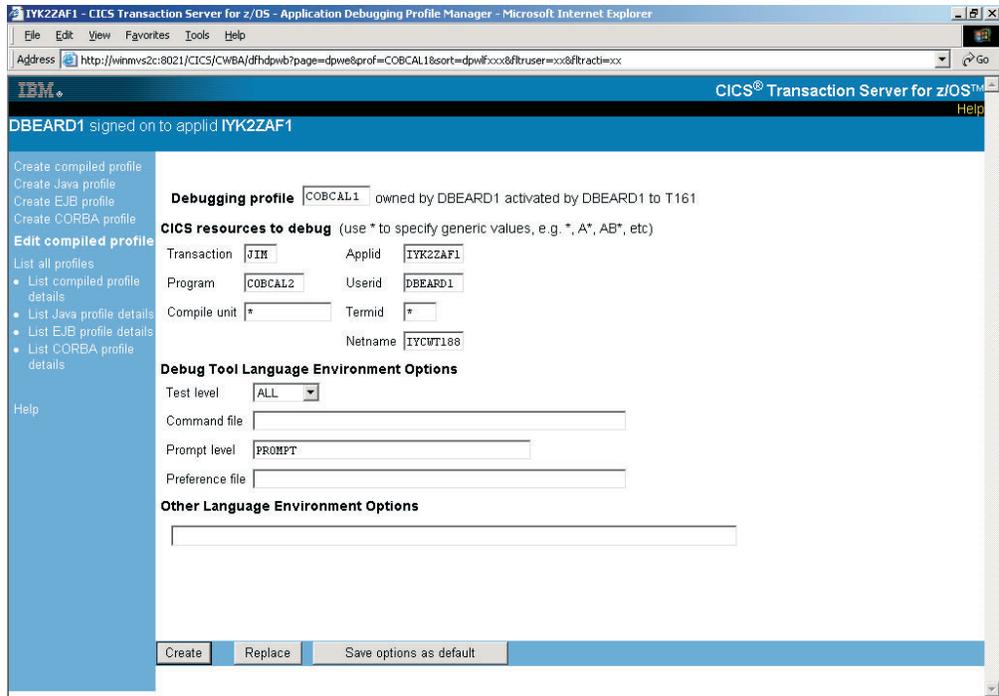


Figure 4. The View compiled profile page

---

## The View Java profile page

Use the “View Java profile” page to view the contents of a debugging profile for compiled language (Language Environment) programs. You cannot change the profile with this page. The information displayed is described in “The Create Java profile page” on page 28.

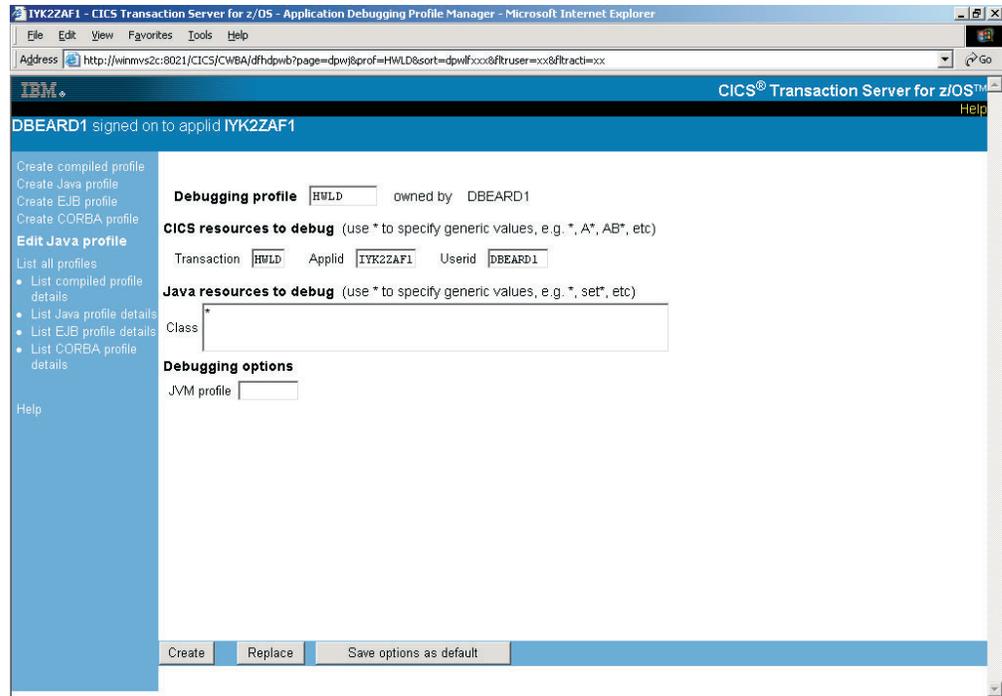


Figure 5. The View Java profile page

## 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 all profiles” page, follow these steps:

1. Click on **Set compiled display device**. CICS displays the “Set compiled 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 all profiles” page.

The “Set compiled 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 (Language Environment) programs when you activate them.

## Specifying the display device when you make a profile active

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

1. Click on **Set compiled display device**. CICS displays the “Set compiled 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 all profiles” page.

CICS will display the “Set compiled display device” page whenever you activate profiles for compiled language (Language Environment) 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 all 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.

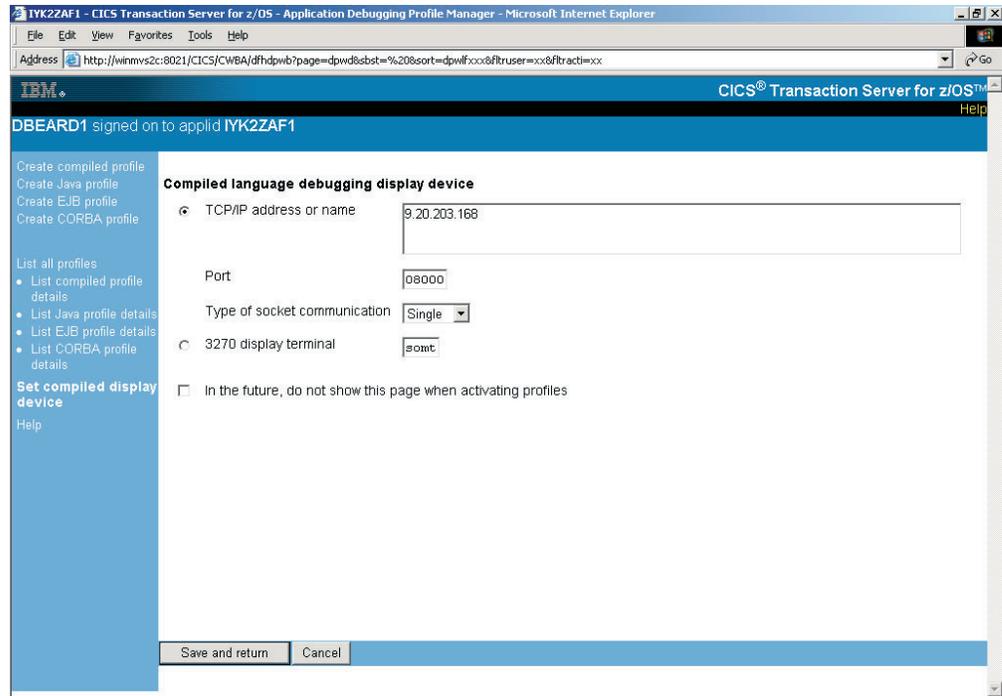


Figure 6. The Set compiled display device page

## 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*
- CICSUSER CORBA  I T*
- $EXAMPLE EJB    I *
- $EXAMPLE JAVA  I TR*
- $EXAMPLE LE1   I T*  P*    *          CICSREG1  PANDREWS  TTT1  Comp
- $EXAMPLE LE2   I TR  *    SAMPCOMPUN + CICSREG2  DRBEARD*  TTT2  Comp
- $EXAMPLE LE3   I TRN3  PROG3  *          CICSREG3  *        TTT2  Comp

7 profile(s). All profiles shown
Enter=Process PF1=Help 2=Filter 3=Exit 4=View 5=Create comp 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
- Comp** 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 Compiled 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 Compiled Debugging Display Device” screen
- PF10 Edit the profile on the line that contains the cursor, using the “Create Compiled 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:

#### To create a profile for a compiled language program

1. Press PF5. CICS displays the “Create Compiled Debugging Profile” screen.
2. Complete the fields that you need to specify your profile.
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.

#### To create a profile for a Java program

1. Press PF6. CICS displays the “Create Java Debugging Profile” screen.
2. Complete the fields that you need to specify your profile. You may 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 40. Before you save the profile, give it a new name.

#### If another user owns the profile

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 40

---

## 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 Compiled Debugging Display Device” screen.

If none of the selected profiles is for a compiled language programs, CICS refreshes the “List Debugging Profiles” screen.

You can choose not to see the “Set Compiled 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 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 Compiled 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 Compiled 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.

**Notes:**

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 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 40. 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. If you do not own the profile, you must make it inactive before you can delete it. To do this, see
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  
Repeat steps 2 through 3 to select all the profiles you want to work with.
4. 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 Compiled Debugging Profile screen

Use the "Create Compiled 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 Compiled 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 Compiled Debugging Profile screen

### Fields on the Create Compiled Debugging Profile screen

The fields on the “Create Compiled Debugging Profile” screen are:

**Create Compiled 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.

#### App1id

# 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 Compiled 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 run-time 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 a debugging profile for a compiled language (Language Environment) program. For more information, see “Specifying default values for Debug Tool and Language Environment 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 dataset 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 dataset or a member of a partitioned data set.

**Other Language Environment options**

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

## Function keys for the Create Compiled Debugging Profile screen

The function keys for the “Create Compiled 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 Language Environment 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 Language Environment 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 Compiled 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
```

Figure 9. The CADP Create Java Debugging Profile screens (Part 1 of 2)

```

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 (Part 2 of 2)

## 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.

#### **Appid**

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 the *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 the *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 Compiled Debugging Profile screen

Use the “View Compiled Debugging Profile” 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 Compiled Debugging Profile LE1      for $EXAMPLE

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

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

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

PF1=Help 3=Exit 12=Return

```

Figure 10. The View Compiled Debugging Profile screen, showing example profile LE1

The information displayed is described in “The Create Compiled Debugging Profile screen” on page 42.

### Function keys for the View Compiled Debugging Profile screen

- The function keys for the “View Compiled 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 Profile” 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

```

Figure 11. The View Java Debugging Profile screen, showing example profile EJB (Part 1 of 2)

```

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 (Part 2 of 2)

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 Profiles” 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 Compiled 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 compiled profile activation” field.
4. Press ENTER. CICS saves the display device settings, and displays the “List Debugging Profiles” screen.

The “Set Compiled 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 Compiled 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 compiled profile activation” field.
4. Press ENTER. CICS saves the display device settings, and displays the “List Debugging Profiles” screen.

CICS will display the “Set Compiled 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 Compiled Debugging Display Device screen

Use the “Set Compiled 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 Compiled 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 compiled profile activation ==>  YES

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

Figure 12. The CADP Set Compiled Debugging Display Device screen

## Fields on the Set Compiled Debugging Display Device screen

The fields on the “Set Compiled 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 compiled profile activation**

Specifies whether you want to display the “Set Compiled Debugging Display Device” screen whenever you activate debugging profiles for compiled language programs:

- YES** Display the “Set Compiled Debugging Display Device” screen whenever debugging profiles are activated. This is the default behavior.
- NO** Do not display the “Set Compiled 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 Compiled Debugging Display Device screen

The function keys for the “Set Compiled 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 *CICS Business Transaction Services* manual.



## 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. For more information about using  
# certificate revocation lists, see the CICS RACF Security Guide.

| 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  
# RACF profile that authorizes CICS to access the certificate revocation lists on the  
# specified 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://cr1.CertificateAuthority.com/CRLList1.cr1  
| http://cr1.CertificateAuthority.com/CRLList2.cr1
```

- | 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:cics311dap  
| http://cr1.CertificateAuthority.com/CRLList1.cr1  
| http://cr1.CertificateAuthority.com/CRLList2.cr1')  
| LENGTH(124) INTERVAL(960000)
```

| This would schedule the CCRL transaction to run in 96 hours. If you enter an invalid  
| URL, you will receive an error message.



## 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 64

---

## Connecting CICS to DBCTL

You can connect CICS to DBCTL by:

1. Typing CDBC CONNect [SUFFIX(xx)], or
2. 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.

You should get the following message:

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

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.)

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:

1. Type  
CDBC DISConnect [IMMediate]

or

2. 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 via 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      : xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.
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:

```
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 re-use 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 the *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 70
- “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. Note that, in this case, the default CRC is the one that must be used.

**Notes:**

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 may need to set the correct AOIS execution parameter. For example, AOIS=S ensures that command authorization is bypassed; with any other value, the command may 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.



---

## 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 may load many definitions in this way, or load none. You may 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 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 the <i>CICS IMS Database Control Guide</i> for details). Leading or embedded blanks are not allowed, but trailing blanks are acceptable. <b>Note:</b> 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.) <b>Note:</b> 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.



**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 on page 73.

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 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:

```
CEBR TSQ AXBYQUEUE1     SYSID CIJP REC      1 OF      3      COL      1 OF      5
ENTER COMMAND ==>
***** TOP OF QUEUE *****
00001 HELLO
00002 HELLO
00003 HELLO
***** 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 47 on page 122) 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.

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

```

CEBR TSQ AXBYQUEUEAME1  SYSID CIJP REC  1 OF  0  COL  1 OF 1 0
ENTER COMMAND ==>
***** TOP OF QUEUE *****
*
***** BOTTOM OF QUEUE *****

TSQUEUE AXBYQUEUEAME1 DOES NOT EXIST
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

```

**Note:** 1 Header 2 Command line 3 Body 4 Message line 5 Menu 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 9.

---

### 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 book, 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 SET commands allow you to query or set:

- Auxiliary trace (“CEBT INQUIRE AUXTRACE” on page 80 and “CEBT SET AUXTRACE” on page 91)
- The dump data set (“CEBT INQUIRE DUMP” on page 82 and “CEBT SET DUMP” on page 93)
- GTF tracing (“CEBT INQUIRE GTFTRACE” on page 83 and “CEBT SET GTFTRACE” on page 94)
- Internal tracing (“CEBT INQUIRE INTTRACE” on page 84 and “CEBT SET INTTRACE” on page 95)
- Surveillance or ‘tracking’ of the active CICS system by the alternate CICS system (“CEBT INQUIRE SURVEILLANCE” on page 85 and “CEBT SET SURVEILLANCE” on page 96)
- Takeover from the active CICS system by the alternate CICS system (“CEBT INQUIRE TAKEOVER” on page 86 and “CEBT SET TAKEOVER” on page 97).

## CEBT INQUIRE AUXTRACE

### CEBT INQUIRE AUXTRACE

#### Function

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).

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

▶—CEBT Inquire Auxtrace—▶

#### Displayed fields

##### A11

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

### CEBT INQUIRE DUMP

#### Function

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

▶▶—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

### Function

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

▶▶—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

### CEBT INQUIRE INTTRACE

#### Function

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

▶▶—CEBT Inquire Intrtrace—▶▶

#### Displayed fields

##### Intrace

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

### Function

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

▶▶—CEBT Inquire Surveillance—▶▶

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

## CEBT INQUIRE TAKEOVER

### CEBT INQUIRE TAKEOVER

#### **Function**

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 PERFORM commands

The CEBT PERFORM commands allow you to:

- Shut down the alternate CICS system (“CEBT PERFORM SHUTDOWN” on page 88)
- Request a dump of the alternate CICS and continue processing (“CEBT PERFORM SNAP” on page 89)
- Cause the alternate CICS to take over from the active CICS (“CEBT PERFORM TAKEOVER” on page 90).

## CEBT PERFORM SHUTDOWN

### CEBT PERFORM SHUTDOWN

#### Function

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

▶▶—CEBT Perform SHUTDOWN—Dump—Immediate————▶▶

#### 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

### Function

Cause a “snapshot” dump of the alternate CICS system to be taken while CICS continues to run.

All CEBT PERFORM SNAP requests result in an SDUMP on an MVS SYS1.DUMPnn 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 SNaP—▶▶

## CEBT PERFORM TAKEOVER

## CEBT PERFORM TAKEOVER

### Function

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

▶▶—CEBT Perform Takeover—Dump————▶▶

### 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.

## 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 140).

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							
▶▶ CEBT Set Auxtrace	Switch ▶▶						
<table border="1"> <tr><td>STArt</td><td>NOswitch</td></tr> <tr><td>Pause</td><td>NExt</td></tr> <tr><td>STOp</td><td>All</td></tr> </table>	STArt	NOswitch	Pause	NExt	STOp	All	
STArt	NOswitch						
Pause	NExt						
STOp	All						

### Options

#### A11

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.

## **CEBT SET AUXTRACE**

### **STAr**t

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.

### **SWi**ch

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

### Function

Change the attributes of a CICS dump data set.

### Description

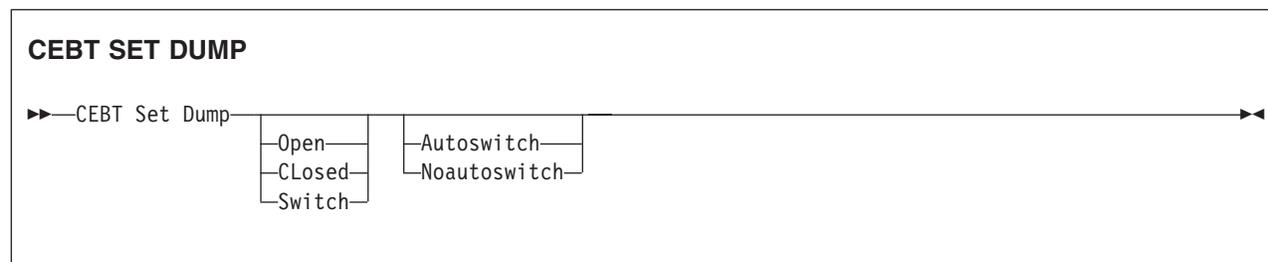
Using CEBT SET DUMP, you can open or close the active CICS dump data set, and specify whether or not the automatic switch to the inactive dump data set is to occur when the active dump data set is full. Instead of using the SET command, you can overwrite your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 140).

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.



### 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

### CEBT SET GTFTRACE

#### Function

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 overwrite 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 ST0 will stop GTF tracing if it is already running.

<b>CEBT SET GTFTRACE</b>	
▶▶—CEBT Set Gtftrace	◀◀
—STArt—	
—STOp—	

#### 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 INTRACE

### Function

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 INTRACE (the minimum abbreviation is CEBT S IN). You get a display that lists the current status, similar to that obtained by CEBT INQUIRE INTRACE. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type CEBT SET INTRACE (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 INTRACE**

▶▶—CEbt Set Intrtrace—▶▶

STArt
STOp

### Options

#### STArt

CICS internal tracing is started.

#### STOp

CICS internal tracing is stopped.

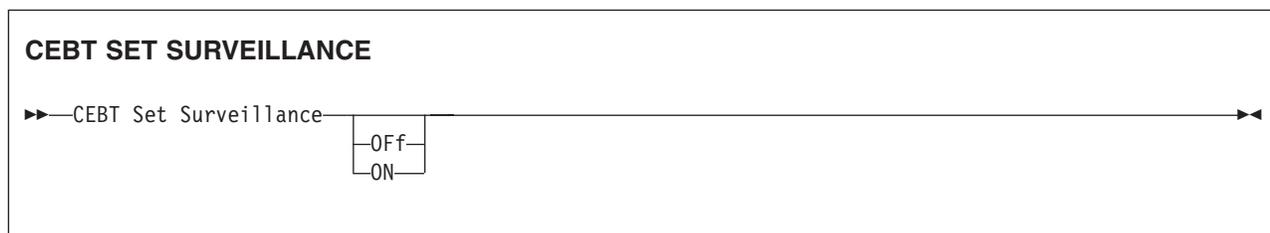
## CEBT SET SURVEILLANCE

## CEBT SET SURVEILLANCE

### Function

Set surveillance (or tracking) of the active CICS by the alternate CICS on or off.

### Syntax



### 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

### Function

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 overwrite 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	
▶▶—CEBT Set Takeover—	◀◀
Automatic	
Command	
Manual	

### 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 SET TAKEOVER

## 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 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    P0Int      R0ute      WAITCics
BUild      EXtract  POP        SEND       WRITe
CAnceL     FEpi     POST       SET         WRITeQ
CHange    FOrmatTImE  PURge     SIGNOff    XctI
COLlect    FREe     PUSh       SIGNON
CONNect    FREEMain  Query     SPOOLClose
CONVerse   GDS      READ       SPOOLOpen
DELAy     GEtmain  READNext   SPOOLRead
DELETE    Handle   READPrev   SPOOLWrite
DELETEQ   IIgnore  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 100 shows the screen displayed for the EXEC CICS RETRIEVE command.

```

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 99. 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.

Start the transaction by entering CEDA on the command line of the screen. Press ENTER. You get the following screen (Figure 30), which shows the options available within CEDA.

```
ENTER ONE OF THE FOLLOWING

Add
ALter
APpend
CHeck
COpy
DEFine
DElete
DISplay
Expand
Install
Lock
Move
REMove
REName
UNlock
USerdefine
View

                                                                    SYSID=CI41 APPLID=IYAHZCCV

PF 1 HELP          3 END          6 CRSR          9 MSG          12 CNCL
```

Figure 30. CEDA transaction: initial screen

CEDA provides the following functions:

- |                |  |
|----------------|--|
| <b>Add</b>     | Adds a group name to a list, and creates a new list if the list did not already exist.       |
| <b>ALter</b>   | Changes some of the attributes of an existing resource definition.                           |
| <b>APpend</b>  | Copies a list to another list.   |
| <b>CHeck</b>   | Checks that the resource definitions within a group or list are consistent with one another. |
| <b>COpy</b>    | Copies one or more resource definitions.   |
| <b>DEFine</b>  | Creates a new resource definition on the CSD file.   |
| <b>DElete</b>  | Erases one or more resource definitions from the CSD file.                                   |
| <b>DISplay</b> | Displays one or more lists or groups on a full-screen panel.                                 |
| <b>Expand</b>  | Shows the resource definitions in one or more groups on a full-screen panel.                 |
| <b>Install</b> | Dynamically makes one or more resource definitions available to an active CICS system.       |

<b>Lock</b>	Assures exclusive access to a group or list by restricting update and delete access to a single operator identifier.
<b>Move</b>	Moves one or more resource definitions from one named group to another.
<b>REMove</b>	Removes a group name from a list.
<b>REName</b>	Renames a resource definition.
<b>UNlock</b>	Removes a lock on a group or list which was previously put there by a CEDA LOCK command.
<b>USerdefine</b>	Instead of using CICS-supplied default values with the DEFINE command, you can set your own defaults and use the USERDEFINE command to create a new resource definition on the CSD file.
<b>View</b>	Lets you look at the resource definition attributes.

This section briefly describes each of these functions and their syntax. For guidance on how to use CEDA, see the *CICS Resource Definition Guide*.

If you enter DEF on the command line and press ENTER, you get the screen shown in Figure 31 which lists the resources:

```

DEF
ENTER ONE OF THE FOLLOWING

Connection      Requestmodel
CORbaserver     Sessions
DB2Conn        TCpipservice
DB2Entry       TDqueue
DB2Tran        Terminal
DJar           TRANClass
DOctemplate    TRANSaction
Enqmodel       TSmodel
File           TYpeterm
Journalmodel   Urimap
Lsrpool       Webservice
Mapset
PARTitionset
PARTner
Pipeline
PROcesstype
PROFile
PROGram

SYSID=JOHN APPLID=I

PF 1 HELP      3 END          6 CRSR          9 MSG          1

```

Figure 31. CEDA transaction: DEFINE screen

If you enter INST on the command line and press ENTER, you get the screen shown in Figure 32 on page 105 which lists the resources:

```

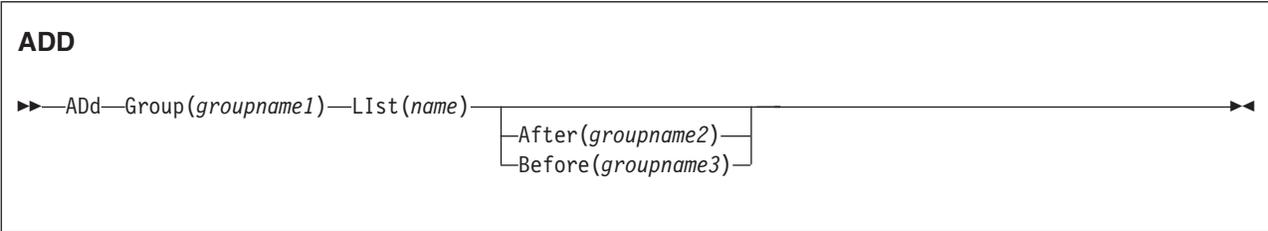
CEDA INST
OVERTYPE TO MODIFY
CEDA Install
All
Connection ==>
CORbaserver ==>
DB2Conn ==>
DB2Entry ==>
DB2Tran ==>
DJar ==>
DOctemplate ==>
Enqmodel ==>
File ==>
Journalmodel ==>
LSrpool ==>
Mapset ==>
PARTitionset ==>
PARTner ==>
Pipeline ==>
PROcesstype ==>
PROfile ==>
PROgram ==>
Requestmodel ==>
Sessions ==>
TCpipservice ==>
TDqueue ==>
TErминаl ==>
+ TRANClass ==>
+ TRANSaction ==>
TSmodel ==>
TYpтерm ==>
Urimap ==>
Webservice ==>
Group ==>
LIst ==>
S GROUP or LIST must be specified.
SYSID=JOHN APPLID=I
PF 1 HELP      3 END      6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 1

```

Figure 32. CEDA transaction: INSTALL screen

## Command syntax

CEDA



## ALTER

▶▶ Alter | Resource | Group(*name*) | attribute list |

## APPEND

▶▶ Append | List(*list1*) | To(*list2*) |

## CHECK

▶▶ Check | Group(*name*) | List(*list1, ..., listn*) | Remotesystem(*sysid*) |

## COPY

▶▶ Copy | AL1 | Resource | Route | REPLACE | MERGE |

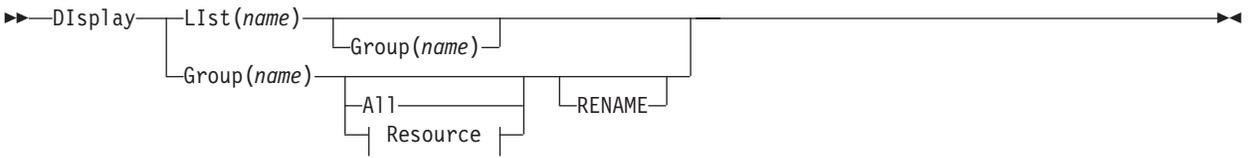
## DEFINE

▶▶ DEfine | Resource | Group(*name*) | attribute list |

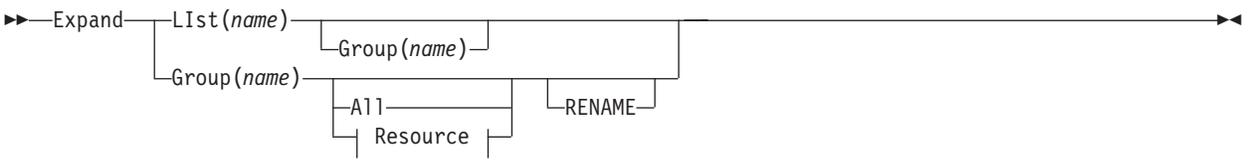
## DELETE

▶▶ DElete | AL1 | Resource | Group(*name*) | REMOVE |

## DISPLAY



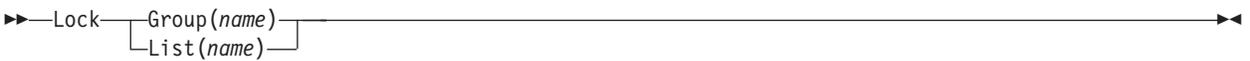
## EXPAND



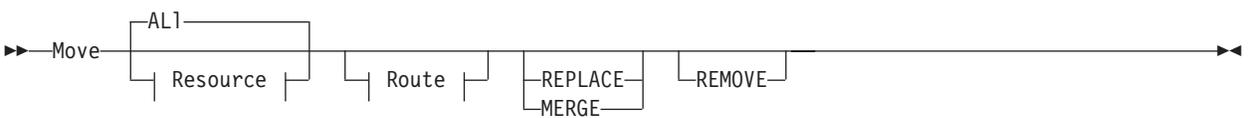
## INSTALL



## LOCK



## MOVE



## REMOVE

▶▶ REMOVE Group(*name*) List(*name*) ▶▶

## RENAME

▶▶ RENAME  
┌── AL1 ──┐  
└── Resource ┘  
┌── Route ┘  
└── REMOVE ┘

▶▶ RENAME Group(*name*) List(*name*) ▶▶

## UNLOCK

▶▶ UNLOCK  
┌── Group(*name*) ┘  
└── List(*name*) ┘

▶▶ UNLOCK Group(*name*) List(*name*) ▶▶

## USERDEFINE

▶▶ USERDEFINE Resource Group(*name*) attribute list ▶▶

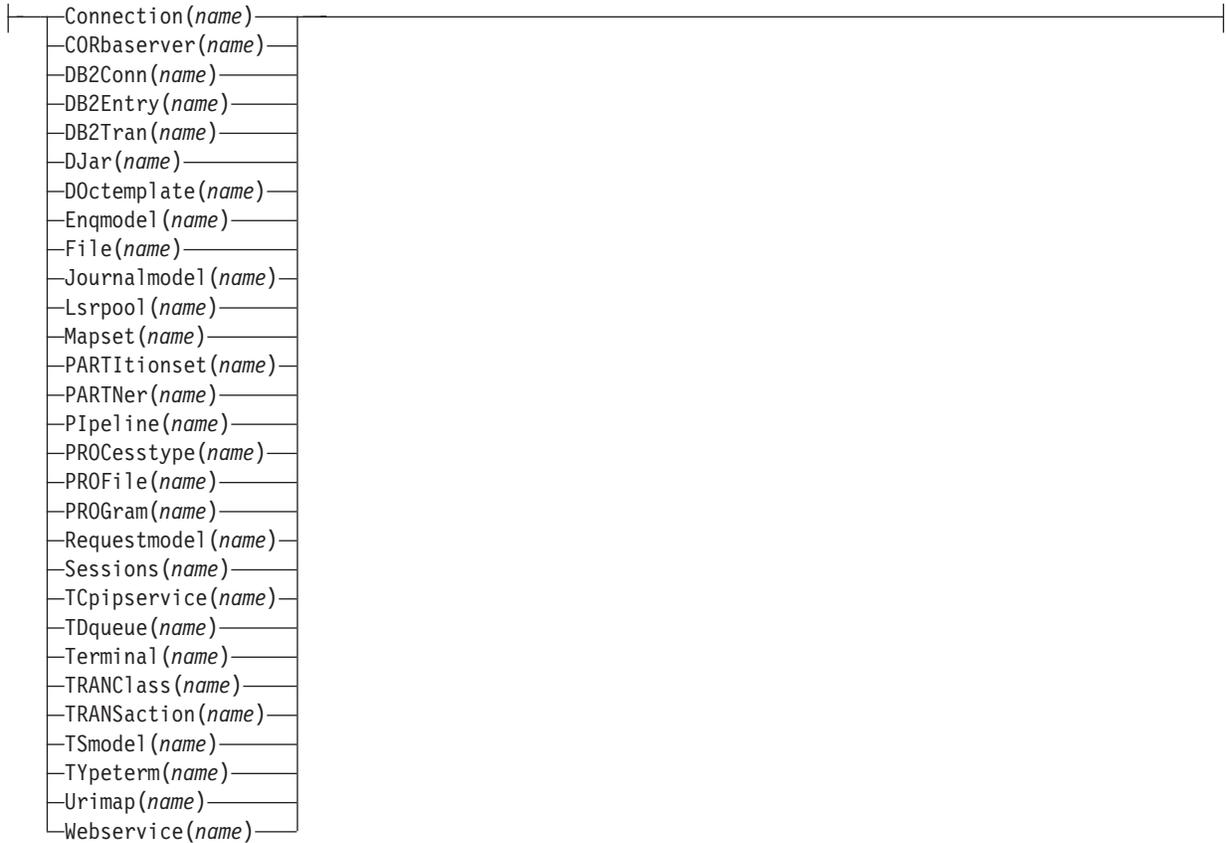
## VIEW

▶▶ VIEW Group(*name*)  
┌── AL1 ──┐  
└── Resource ┘

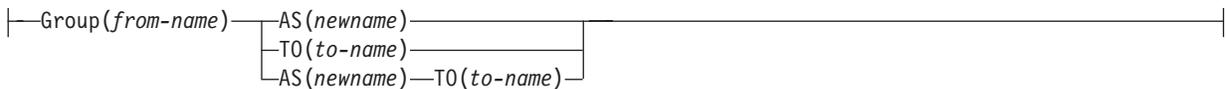
▶▶ VIEW Group(*name*) ▶▶

## CEDA continued

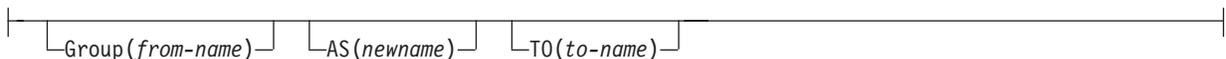
### Resource:



### Route (Copy and Move):



### Route (Rename):



### attribute list:

for details consult the *Resource Definition Guide*

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 that 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 8 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 *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. This 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 on 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

### CSD compatibility panel

Figure 33 shows a normal ALTER panel for the bottom part of the REQUESTMODEL definition, with the OMGINTEGER, OMGMODULE, and OMGOPERATION attributes, that are now obsolete, protected as indicated by the colon symbol (:).

```
ALTER                                     RELEASE 0640
OVERTYPE TO MODIFY
.
.
.
TRANSACTION ATTRIBUTES
  TRansid      ==> CIRP

PREVIOUS ATTRIBUTES
  OMGModule    :
  OMGInterface :
  OMGOperation :
```

Figure 33. 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
Beaname    ==> *
           ==>
           ==>
           ==>
INTFacetype ==> Both          Both | Home | Remote
CORBA PARAMETERS
Module     :
           :

```

Figure 34. 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 :
OMGInterf :
OMGOperat :

```

Figure 35. The DEFINE panel for REQUESTMODEL

See the *CICS Resource Definition Guide* for details of using CEDA. See the the *CICS Resource Definition Guide* for background information about the CSD compatibility panel.

Use the PF2 key to change to the compatibility mode panel shown in Figure 36 on page 114. 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 36. The ALTER panel for REQUESTMODEL: compatibility 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 CICS Transaction Server for z/OS, Version 2 Release 1 attribute values.

**Note:** If you migrate a CICS Transaction Server for OS/390, Version 1 Release 3 request model resource definition to CICS Transaction Server for z/OS, Version 2 Release 1, you must remove the CICS Transaction Server for OS/390, Version 1 Release 3 attribute values and enter CICS Transaction Server for z/OS, Version 2 Release 1 attribute values. You cannot install a CICS Transaction Server for OS/390, Version 1 Release 3 request model definition in a CICS Transaction Server for z/OS, Version 2 Release 1 system.

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 37. 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 38. 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 39. 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 40. 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 overwrite. 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 41. 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 42. 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 43. 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 44. 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.

### 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.

```
ENTER ONE OF THE FOLLOWING

Add
ALter
APpend
ChEck
COpy
DEFine
DELeTe
DIisplay
EXpand
Lock
Move
REMove
REName
UNlock
USerdefine
View

                                     SYSID=CI41 APPLID=IYAHZCCV

PF 1 HELP      3 END      6 CRSR      9 MSG      12 CNCL
```

Figure 45. CEDB transaction: initial screen

## 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.

```
ENTER ONE OF THE FOLLOWING

Display
Expand
View

                                     SYSID=CI41 APPLID=IYAHZCCV

PF 1 HELP      3 END      6 CRSR      9 MSG      12 CNCL
```

Figure 46. CEDC transaction: initial screen

For background information about the CEDA, CEDB, and CEDC transactions, see the *CICS Resource Definition Guide*.

---

## Chapter 12. CEDF & 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 transeive (ATI/TTI) status.

### Command syntax



### 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
```

## 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 47 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'
EIBFN        = X'0000'
EIBRCODE     = X'000000000000'
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 47. 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

See Figure 48 on page 124 for an example of the screen from which CEBR and CECI can be invoked.

**Note:** 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.

```
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 DISPLAY
PF7 : SCROLL BACK HALF  PF8 : SCROLL FORWARD HALF PF9 : UNDEFINED
PF10: SCROLL BACK FULL  PF11: SCROLL FORWARD FULL PF12: REMEMBER DISPLAY
```

Figure 48. Typical EDF display from which CEBR or CECI can be invoked

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; for detailed guidance on using them, see the *CICS Application Programming Guide*.)

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 99; for detailed guidance on using them, 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.

## Command syntax



## 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
```

### **trandid**

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 *trandid* 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.

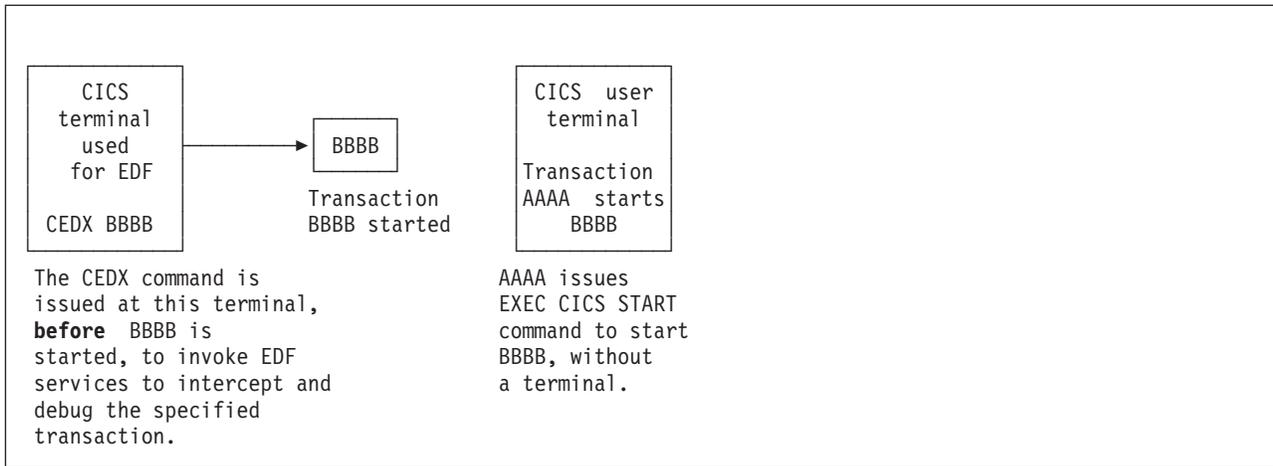


Figure 49. Invoking EDF by CEDX

---

## 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.

### Notes:

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). Starting a transaction at the system console is described in *CICS Supplied Transactions*

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 book, the minimum permitted abbreviation is given in uppercase characters, and the rest of the keyword in lowercase characters.

## CEKL INQUIRE TASK

### Function

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.

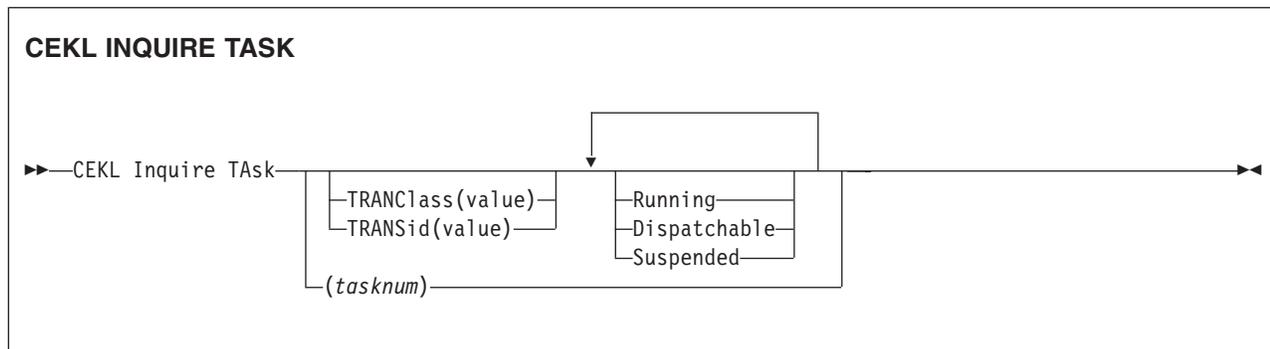
From a system console enter:

```
/F cicsregionname,CEKL INQUIRE TASK(tasknum)
```

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:



### 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(aaaaaaaaa) 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**(*ttttttt*)

is the name of the transaction class (up to 8 characters) associated with the task. If the task is not defined in a class then this is blank.

**USE**(*aaaaaaaa*)

displays the userid for the task.

## CEKL INQUIRE 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(rrrrrrr)**

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 MEAC022X,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 L001 USE(CICSUSER) ATT(00033) CPU(00023) RUA(00010)

```

Figure 50. CEKL INQUIRE TASK output

The second example shows where a task number has been selected and one record is returned.

```

F MEAC022X,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 51. CEKL INQUIRE TASK output

---

## CEKL SET TASK

### Function

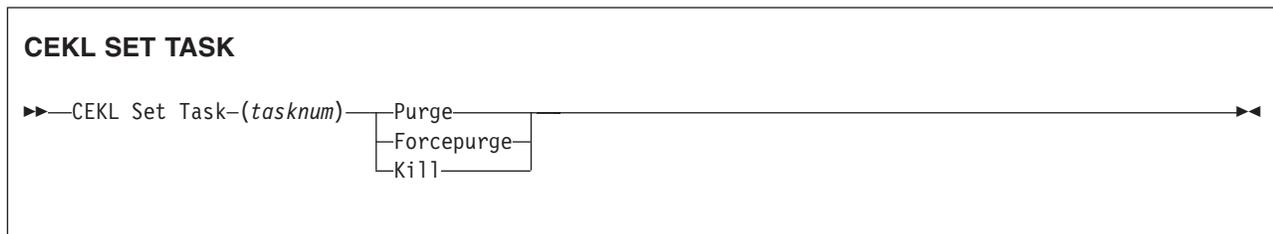
Requests the cancellation of a user task.

### Description

CEKL SET TASK initiates a PURGE, FORCEPURGE, or KILL request.

### Syntax

The syntax of the CEKL SET TASK command is as follows:



### 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 MEAC022X,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 MEAC022X,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 MEAC022X,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 MEAC022X,CEKL I TA(62)
+IYK2ZFX3 DFHCQ0242I CEKL INQUIRE: no tasks matched selection criteria.

```

Figure 52. CEKL SET TASK output

## CEKL SET TASK

---

## Chapter 14. 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 page 79)
- CEOT (see page 521)
- CEST (see page 531).

The CECL transaction (see page 99) ) 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 9. 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.

Enter CEMT on the command line of the screen, as follows:

CEMT

Press the ENTER key. You get the following response:

```
STATUS:  ENTER ONE OF THE FOLLOWING

Discard
Inquire
Perform
Set
```

Figure 53. CEMT transaction: initial screen

This display prompts you to enter one of the keywords shown.

If you then type:

D

and press ENTER, you get the following screen:

```
D
STATUS:  ENTER ONE OF THE FOLLOWING

Autinstmodel PROCesstype
CONnection   PROFile
CORbaserver  PROGram
DB2Conn      Requestmodel
DB2Entry     TClass
DB2Tran      TCPipservice
DJar         TDqueue
DOctemplate  TERminal
Enqmodel     TRansaction
FENode       TSmodel
FEPOol       Urimap
FEPropset
FETarget
File
JModel
JJournalname
PARTner
  S At least 4 characters of DISCARD must be specified.
                                                    SYSID=JOHN APPLID=I

PF 1 HELP      3 END      5 VAR      9 MSG
```

Figure 54. CEMT transaction: DISCARD screen

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      JVM          TAsk         WEBSservice
AUTOinstall  DUmPds        JVMPool     TClass      WOrkrequest
AUXtrace     ENQ           Line        TCPIP
BEan         ENQModel     MODename    TCPIPService
BRfacility    EXci         MONitor     TDqueue
CFdtpool     FEConnection Netname     TErminAl
Classcache   FENode       PArtner    TRAnSACTION
CONnection   FEPOol       PiPeline   TRDumpcode
CORbaserver  FEPRopset    PROCesstype TSMODEL
DB2Conn      FETarget     PRoFile    TSPool
DB2Entry     File         PRoGRAM    TSQueue
DB2Tran      Gtftrace     REquestmodel UOW
DEletshipped Host          RRms       UOWDsnfail
DISpatcher   INttrace     STAtistics UOWLink
DJar         IRc          STReamname URimap
DOctemplate JModel       SYDumpcode Vtam
DSAs         JOurnalname SYStem     WEb

```

Figure 55. CEMT transaction: INQUIRE screen

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 56. CEMT transaction: PERFORM screen

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  FEPool      STatistics  WEBSERVICE
AUXtrace     FETarget    SYDumpcode  Workrequest
Brfacility     FILE        SYSTEM
Classcache   Gtftrace    TAsk
CONnection   Host        TClass
CORbaserver  INttrace    TCPIP
DB2Conn      IRc         TCPIPService
DB2Entry     JOurnalname TDqueue
DB2Tran      JVM         TERminal
DEletshipped JVPool      TRAnsaction
DISpatcher   Line        TRDumpcode
DSAs         MODename    TSqueue
DSName       MONitor     UOW
DUmpds       Netname     UOWLink
Enqmodel     PIPEline    URimap
FEConnection PROCesstype Vtam
FENode       PROGram     WEb

PF 1 HELP      3 END        5 VAR        9 MSG

```

Figure 57. 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 305.

## 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 142).
- 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 information about TLT, XLT, and PLT, 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(T01,T02,T03,T04) 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 58.

```

?INQ TERM
STATUS:  COMMAND SYNTAX CHECK
CEMT Inquire TErминаl()
 < CLass() | AL1 >
 < TRansaction() >
 < PRiority() >
 < PAgeable | AUtopageable >
 < Inservice | Outservice >
 < ATi | NOAti >
 < TTi | NOTTi >
 < NEtname() >
 < ACquired | REleased >
 < CReate | NOCreate >
 < REMotesystem() >

```

Figure 58. 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 59.

```

in task
STATUS:  RESULTS - OVERTYPE TO MODIFY

*** Sample screen required ***

```

Figure 59. 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 60 on page 143).

```

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 60. Terminal settings for a selected terminal

Overtime 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' ('%c' 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; for programming information, see the *CICS System Programming Reference*.<sup>1</sup>

## 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 the *CICS RACF Security Guide*. CEMT calls the exec interface programs to process the user's commands, in much the same way as CECL 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*.

---

1. Users of earlier releases of CICS may be familiar with the old programmable interface to the master terminal program (DFHEMTA). Its use is still supported, though the documentation is available only in the CICS libraries for releases prior to CICS Transaction Server for z/OS, Version 3 Release 1.

---

## 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(resource name). 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
- CorbaServer
- DB2 connection
- DB2 entry
- DB2 transaction
- Deployed JAR file
- Document template
- Enqueue model
- FEPI node
- FEPI pool
- FEPI property set
- FEPI target
- File
- Journal model
- Journal name
- 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(resource name)
```
2. If you do not know the name of the resource, type:  

```
CEMT INQUIRE resourcetype
```

and a screen similar to Figure 61 on page 146 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.

## CEMT DISCARD commands

```
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)
```

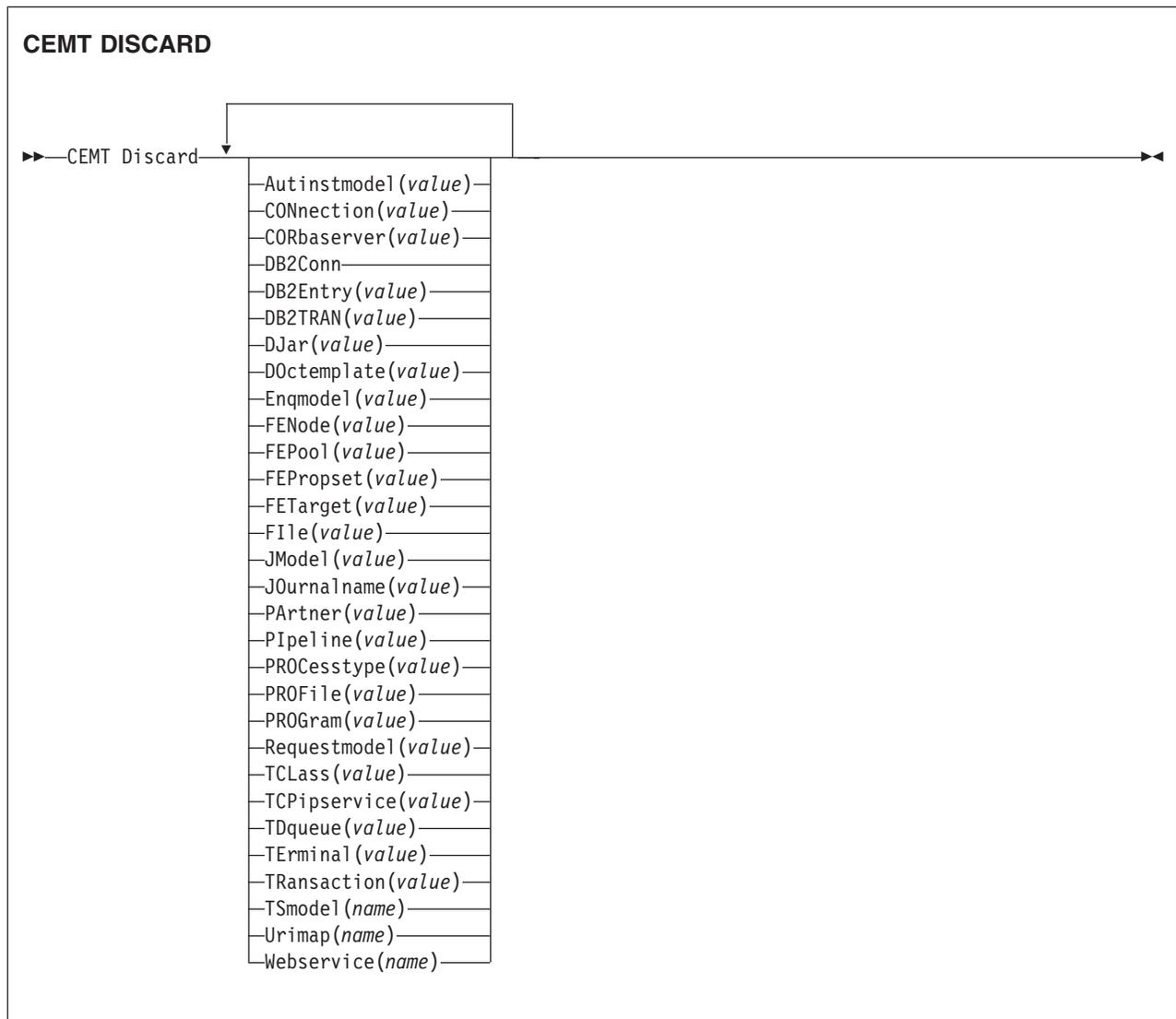
Figure 61. Example of screen showing the discarding of resources

### Notes:

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



## 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.

## CEMT DISCARD commands

- 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 homes of enterprise beans in the deployed JAR file are not retracted from the namespace because other AORs might still have the homes installed. However, before discarding a deployed JAR file from the last AOR in which it is installed, you might wish to issue a PERFORM DJAR RETRACT command to retract references to home interfaces.

**D0ctemplate**(*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**

This is described in the *CICS Front End Programming Interface User's Guide*.

**FEPool**

This is described in the *CICS Front End Programming Interface User's Guide*.

**FEPropset**

This is described in the *CICS Front End Programming Interface User's Guide*.

**FETarget**

This is described in the *CICS Front End Programming Interface User's Guide*.

**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.

**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 DFHJnn, 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.

**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.

## CEMT DISCARD commands

### **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 TCP/IP 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.

### **TErминаl**(*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**(*value*)

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**(*value*)

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**(*value*)  
| 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 391.

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 411.

If you enter INQUIRE, you get the following display:

```

INQ                                     ←Command Line
STATUS:  ENTER ONE OF THE FOLLOWING OR HIT ENTER FOR DEFAULT ←Status Line

AUTInstmodel DSName      JVM          TAsk          WEBSservice
AUTOinstall  DUMPds      JVMpool    TClass        Workrequest
AUXtrace     ENQ          Line       TCPIP
BEan         ENQModel  MODename  TCPIPService
BRfacility    EXci       MONitor   TDqueue
CFdtpool     FEConnection Netname    TErminAl
CLasscache   FENode     PArtner   TRAnSACTION
CONnection   FEPOol     PIPeline  TRDumpcode
CORbaserver  FEPropset  PROCesstype TSMoDel
DB2Conn      FETarget   PROFile   TSPool
DB2Entry     File       PROGram   TSQueue
DB2Tran      Gtftrace   REquestmodel UOW
DEletshipped Host        RRms      UOWDsnfail
DISpatcher   INttrace   STATistics UOWLink
DJar         IRc        STReamname URimap
DOctemplate  JModel     SYDumpcode Vtam
DSAs         JOurnalname SYSTEM     WEB

PF 1 HELP      3 END      5 VAR          SYSID=CICS APPLID=CICSHTC1
9 MSG          ←PF keys
    
```

Figure 62. Sample of the screen following the INQUIRE command

You can inquire about any of the displayed options by typing its keyword after INQUIRE on the command line. For example,  
 INQUIRE PROGRAM

## CEMT INQUIRE COMMANDS

gives you the status of all programs, and for each program gives its attributes. Full details are given in “CEMT INQUIRE PROGRAM” on page 283.

# CEMT INQUIRE AUTINSTMODEL

### Function

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(dfh1u0e2)`), the names of all the autoinstall models will be displayed.

### Sample screen

```
IN AUTIN
STATUS: RESULTS
Aut i (DFHLU0E2)
Aut i (DFHLU0M2)
Aut i (DFHLU0M3)
Aut i (DFHLU0M4)
Aut i (DFHLU0M5)
Aut i (DFHLU2 )
Aut i (DFHLU2E2)
Aut i (DFHLU2E3)
Aut i (DFHLU2E4)
Aut i (DFHLU2M2)
Aut i (DFHLU2M3)
Aut i (DFHLU2M4)
Aut i (DFHLU2M5)
Aut i (DFHLU3 )
Aut i (DFHLU62T)
Aut i (DFHSCSP )
Aut i (DFH3270 )
+ Aut i (DFH3270P)
```

Figure 63. 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**

▶▶—CEMT Inquire AUTInstmodel—▶▶  
└─A11─┘

**Displayed fields**

A11

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

## CEMT INQUIRE AUTOINSTALL

## CEMT INQUIRE AUTOINSTALL

### Function

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 *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 overwrite them as required.

### Sample screen

```
INQ AUTOINSTALL
STATUS: RESULTS - OVERTYPE TO MODIFY
Auto Max( 100 ) Pro( DFHZATDX ) Cur(000) Pro Ena
```

Figure 64. CEMT INQUIRE AUTOINSTALL screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 65.

```
IN AUTOIN
RESULT - OVERTYPE TO MODIFY
Autoinstall
Maxreqs( 100 )
Program( DFHZATDX )
Curreqs( 000 )
Consoles( Fullauto )
Enablestatus( Enabled )
Aibridge( Autotermid )
```

Figure 65. The expanded display of an individual entry

## CEMT INQUIRE AUTOINSTALL

▶—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:

- |                 |  |
|-----------------|--|
| <b>Fullauto</b> | 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. |
| <b>Progauto</b> | Autoinstall for consoles is active, with the support of the autoinstall control program (specified on the AEXIT system initialization parameter). The autoinstall control program is required to provide the terminal ID and, optionally, a delete-delay period.     |
| <b>Noauto</b>   | 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:

- |                  |   |                  |   |                 |   |
|------------------|---|------------------|---|-----------------|---|
| <b>Enabled</b>   | 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: <table border="0" style="margin-left: 20px;"> <tr> <td style="vertical-align: top;"><b>Terminals</b></td> <td>MAXREQS not equal 0 and autoinstall control program is enabled.</td> </tr> <tr> <td style="vertical-align: top;"><b>Consoles</b></td> <td> <ol style="list-style-type: none"> <li>1. CONSOLES displays FULLAUTO.</li> <li>2. CONSOLES displays PROGAUTO and the autoinstall control program is enabled.</li> </ol> </td> </tr> </table> | <b>Terminals</b> | MAXREQS not equal 0 and autoinstall control program is enabled. | <b>Consoles</b> | <ol style="list-style-type: none"> <li>1. CONSOLES displays FULLAUTO.</li> <li>2. CONSOLES displays PROGAUTO and the autoinstall control program is enabled.</li> </ol> |
| <b>Terminals</b> | MAXREQS not equal 0 and autoinstall control program is enabled.   |                  |   |                 |   |
| <b>Consoles</b>  | <ol style="list-style-type: none"> <li>1. CONSOLES displays FULLAUTO.</li> <li>2. CONSOLES displays PROGAUTO and the autoinstall control program is enabled.</li> </ol>   |                  |   |                 |   |
| <b>Disabled</b>  | Neither consoles nor terminals can be autoinstalled in CICS. DISABLED is returned for the following conditions: <table border="0" style="margin-left: 20px;"> <tr> <td style="vertical-align: top;"><b>Terminals</b></td> <td>MAXREQS equal 0, or autoinstall control program is disabled.</td> </tr> <tr> <td style="vertical-align: top;"><b>Consoles</b></td> <td> <ol style="list-style-type: none"> <li>1. CONSOLES displays NOAUTO.</li> <li>2. CONSOLES displays PROGAUTO but the autoinstall control program is disabled.</li> </ol> </td> </tr> </table>   | <b>Terminals</b> | MAXREQS equal 0, or autoinstall control program is disabled.    | <b>Consoles</b> | <ol style="list-style-type: none"> <li>1. CONSOLES displays NOAUTO.</li> <li>2. CONSOLES displays PROGAUTO but the autoinstall control program is disabled.</li> </ol>  |
| <b>Terminals</b> | MAXREQS equal 0, or autoinstall control program is disabled.  |                  |   |                 |   |
| <b>Consoles</b>  | <ol style="list-style-type: none"> <li>1. CONSOLES displays NOAUTO.</li> <li>2. CONSOLES displays PROGAUTO but the autoinstall control program is disabled.</li> </ol>  |                  |   |                 |   |

**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.)

## CEMT INQUIRE 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 140).

For information on using auxiliary tracing, see the *CICS Problem Determination Guide*. See also the description of the CETR transaction, Chapter 19, "CETR—trace control," on page 533.

### 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 66. 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 67 on page 160.

## CEMT INQUIRE AUXTRACE

```
I AUX
RESULT - OVERTYPE TO MODIFY
Auxtrace
Curaxds(A)
Status( Stop )
Switchstatus( Noswitch )
Switchaction(      )
```

Figure 67. The expanded display of an individual entry

### CEMT INQUIRE AUXTRACE

▶—CEMT Inquire AUXtrace—▶

#### Displayed fields

##### Auxiliary

identifies this panel as relating to auxiliary trace.

##### Curaxds(*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:

- |              |   |
|--------------|---|
| <b>Start</b> | Auxiliary tracing is in progress (the auxiliary trace data set is open).  |
| <b>Pause</b> | 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. |
| <b>Stop</b>  | 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:

- |               |  |
|---------------|--|
| <b>Switch</b> | 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:

- |                 |  |
|-----------------|--|
| <b>Noswitch</b> | Switching does not occur without operator intervention.  |
| <b>NEXT</b>     | Automatic switching occurs when the current trace data set is full, but not subsequently.                              |
| <b>ALI</b>      | 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

### Function

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 68. 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 69. The expanded display of an individual entry

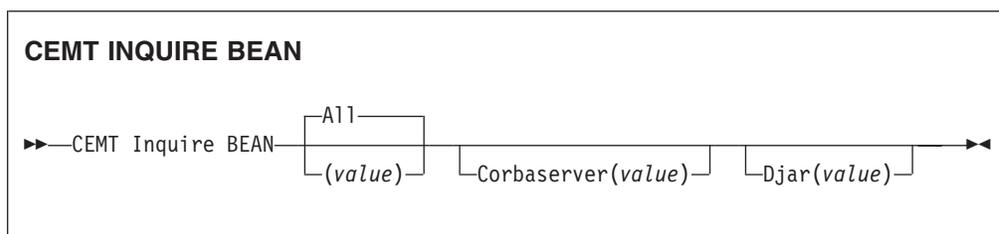
If you inquire against a specific bean and press ENTER, CICS displays an expanded format.

## CEMT INQUIRE BEAN

```
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 70. The expanded display of a specific entry



### Displayed fields

#### A11

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

### Function

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() >
  < Keetime() >
  < RELeased >
  < ACquired | AVailable >
  < LINKSYSTEM() >
  < LINKSYSNet() >
  < REMOTESYSTEM() >
  < REMOTESYSNet() >
  < L0cal | Shared >
```

Figure 71. CEMT INQUIRE BRFACILITY syntax screen

To change various attributes, you can:

- Overtyping your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 140).
- Use the CEMT SET BRFACILITY command.

## CEMT INQUIRE BRFACILITY

### 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
```

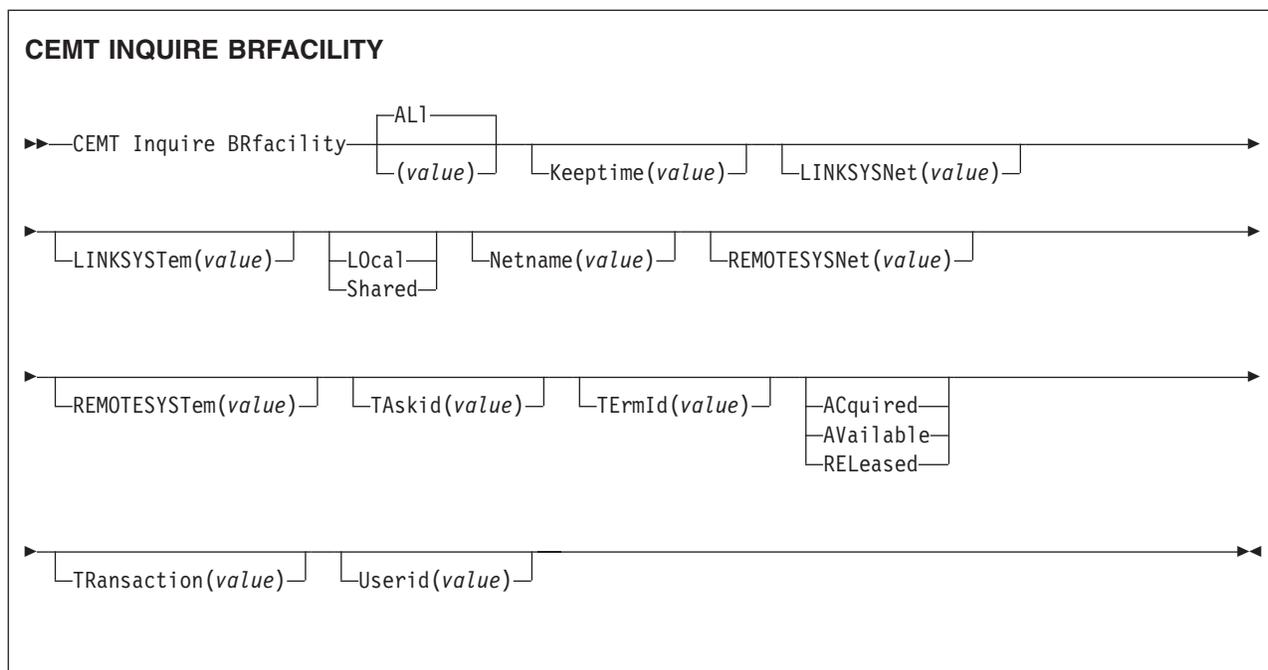
Figure 72. CEMT INQUIRE BRFACILITY screen

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)
Keptime(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 73. The expanded display of an individual bridge facility entry



## Displayed fields

*(value)*

is the 8-byte bridge facility token of a specific bridge facility to be displayed.

**A11**

is the default. All bridge facilities are displayed.

**Keetime** *(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 keep time specified when the facility was allocated,
  - or a default value of 5 minutes if no keep time was specified,
 if the keep time, so determined, is larger than the value of SIT parameter BRMAXKEEPTIME, it is reduced to BRMAXKEEPTIME.
- If the facility being displayed is a Web-Bridge, the keep time is initially set to be the Webdelay 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.

## CEMT INQUIRE BRFACILITY

### 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

### Function

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

```

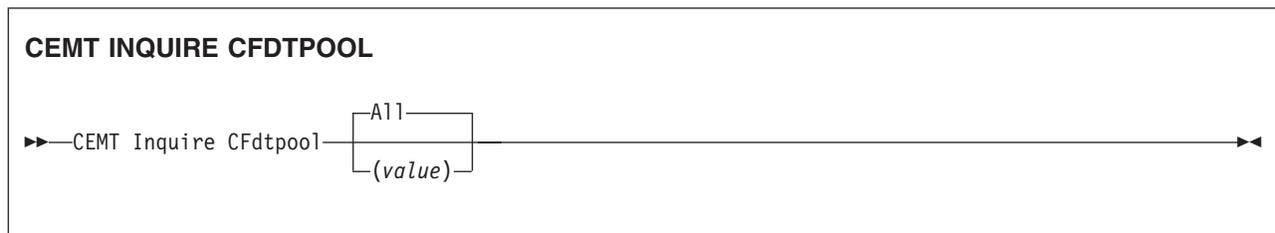
inquire cfdt
STATUS: RESULTS
Cfd(DTPOOL1 ) Con

RESPONSE: NORMAL
PF 1 HELP      3 END      5 VAR      7 SBH 8 SFH 9 MSG 10 SB 11 SF

SYSID=HA61 APPLID=CICSHA61
TIME: 15.54.22 DATE: 09.03.98

```

Figure 74. CEMT INQUIRE CFDTPOOL screen



### Displayed fields

#### A11

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:

## CEMT INQUIRE CFDTPOOL

- 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

### Function

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 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.
- 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.

### Sample screen

```
i classcache
STATUS: RESULTS - OVERTYPE TO MODIFY
Cla Ena Sta Pro(DFHJVMCC) Dat(10/14/03) Tim(16:57:24)
Tot(0002) 01d(0000) Pha(0000) Res Caches(50M)
```

Figure 75. CEMT INQUIRE CLASSCACHE screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 76 on page 170.

## CEMT INQUIRE CLASSCACHE

```
i classcache
RESULT - OVERTYPE TO MODIFY
Classcache
Autostartst( Enabled )
Status(Started)
Profile(DFHJVMCC)
Datestarted(10/14/03)
Timestarted(16:57:24)
Totaljvms(0002)
Oldcaches(0000)
Phasingout(0000)
Reusest(Reset)
Cachesize(50M          )
Cachefree(45260539    )
```

Figure 76. The expanded display for CEMT INQUIRE CLASSCACHE

### INQUIRE CLASSCACHE

▶—CEMT INQUIRE CLASSCACHE—▶

### Displayed fields

#### Autostartst

displays the status of autostart for the shared class cache. The values are:

##### Enabled

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

##### Disabled

If the shared class cache has been stopped or has not yet been started on this CICS execution, an explicit CEMT PERFORM CLASSCACHE Start command is required to start it. If the status of the shared class cache is Stopped and autostart is disabled, and CICS receives a request to run a Java application in a JVM whose profile requires the use of the shared class cache, the request fails.

#### Cachefree(*value*)

displays the amount of free space in the shared class cache, in bytes.

#### 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 initialization 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 worker JVMs that are dependent on them to be phased out. If the status of the current shared class cache is Stopped, and worker JVMs are still dependent on it, then that shared class cache is included in the number of old shared class caches.

**Phasingout**(*value*)

displays the number of worker JVMs that are dependent on an old shared class cache, and are being phased out. If the status of the current shared class cache is Stopped, then any worker JVMs that are still dependent on it are included in the number of worker JVMs being phased out.

**Profile**(*value*)

If the status of the shared class cache is Stopped, this displays the eight-character name of the JVM profile that will be used for a master JVM to start the shared class cache. If the status of the shared class cache is Started, Starting or Reloading, this displays the eight-character name of the JVM profile that was used for the last valid request to start or reload the shared class cache. This name is displayed even if the shared class cache fails to start or reload. The displayed JVM profile is used next time you issue the command to start or reload the shared class cache, unless you specify a different JVM profile using the Profile option.

**Reusest**

displays the level of reusability for the master JVM that initializes the shared class cache. All the worker JVMs in a CICS region inherit their level of reusability from the master JVM. The values are:

**Reset** The master JVM and worker JVMs are resettable.

**Reuse** The master JVM and worker JVMs are continuous.

**Unknown**

The level of reusability for the master JVM and worker JVMs is not known, because the shared class cache is not started.

**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 worker JVMs. This value in the CEMT display includes both the status STARTED and the transient status RELOADING, which occurs when a CEMT PERFORM CLASSCACHE Reload command has been issued, and a new shared class cache is being loaded to replace the existing shared class cache. While the shared class cache is reloading, worker JVMs, both those that were already allocated to tasks and those that were allocated to tasks after the command was issued, continue to use the existing shared class cache until the new shared class cache is ready.

**Stopped**

The shared class cache has either not been initialized on this CICS execution, or it has been stopped by a CEMT PERFORM CLASSCACHE Phaseout, Purge or Forcepurge or CEMT SET JVMPOOL Phaseout, Purge or Forcepurge command. If autostart is disabled, requests to run a Java application in a JVM whose profile requires the use of the shared class cache (that is, requests for worker JVMs) will fail. If autostart is enabled, a new shared class cache will be initialized as soon as CICS receives a request to run a Java application

## CEMT INQUIRE CLASSCACHE

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, either through the autostart facility or because an explicit CEMT PERFORM CLASSCACHE Start command was issued. While the shared class cache is starting, worker 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 worker JVMs 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 worker JVMs in the CICS region that are dependent on a shared class cache. This includes both the worker JVMs that are dependent on the current shared class cache, and any worker JVMs that are dependent on an old shared class cache and are being phased out.

## CEMT INQUIRE CONNECTION

### Function

Retrieve information about system connections.

### Description

CEMT INQUIRE CONNECTION returns information about the status of connections to a remote system or to another CICS region.

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.

**A11**

is the default.

## CEMT INQUIRE CONNECTION

### 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(MR02) Net(CICSHAA1)  Ins Acq Irc          Unk
Con(MR04) Net(CICSHAA2)  Ins Acq Irc          Unk
Con(MR05) Net(CICSHAA3)  Ins Acq Irc          Unk
Con(MR06) Net(CICSHAB1)  Ins Acq Xcf
Con(VTA1) Net(IYCQTC03)  Ins Acq Vta Appc
                                     Nqn(GBIBMIYA.IYCQTC03)
```

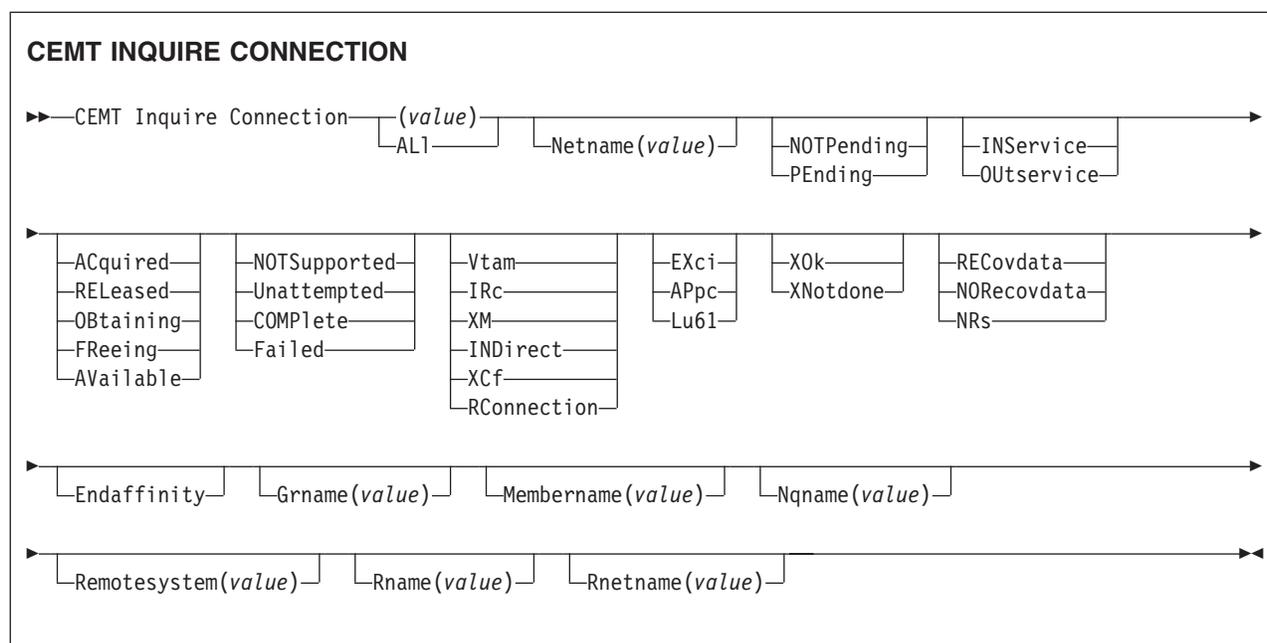
Figure 77. CEMT INQUIRE CONNECTION screen

**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 78.

```
IN CONN
RESULT - OVERTYPE TO MODIFY
Connection(MR05)
Netname(CICSHAA3)
Pendstatus( Notpending )
Servstatus( Inservice )
Connstatus( Acquired )
Accessmethod(Irc)
Purgetype(          )
Protocol()
Xlnstatus()
Recovstatus( Norecovdata )
Uowaction(          )
Cqp( Notsupported )
Grname()
Memername()
Affinity(          )
Nqname()
Remotesystem()
Rname()
Rnetname()
```

Figure 78. The expanded display of an individual entry



## 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 MVS sysplex. It is selected dynamically by CICS for such links when the access method is defined as IRC or XM in the connection definition.
- 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

## CEMT INQUIRE CONNECTION

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 successfully when the connection was release. The status reverts to UNATTEMPTED when the connection is re-acquired.
- Failed** The connection quiesce protocol failed (for instance, because 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-CICS Transaction Server for z/OS 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.

## CEMT INQUIRE CONNECTION

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.

#### **Cancel**

In some extreme cases (for example, if an error occurs during backout processing), CICS might terminate abnormally. 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

Table 2. System AIDs requiring FCANCEL to remove them (continued)

Description	Tranid
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.

### **Fcancel**

All AIDs, including system AIDs, queuing for the specified connection are to be canceled. See Table 2 on page 178 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.

## CEMT INQUIRE CONNECTION

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

## CEMT INQUIRE CONNECTION

- Does not support synchronization level 2 conversations.

For information about the APPC exchange log names process, see the *CICS Intercommunication Guide*.

## CEMT INQUIRE CORBASERVER

## CEMT INQUIRE CORBASERVER

### Function

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 )          Ssslun(IIOPSSLU)

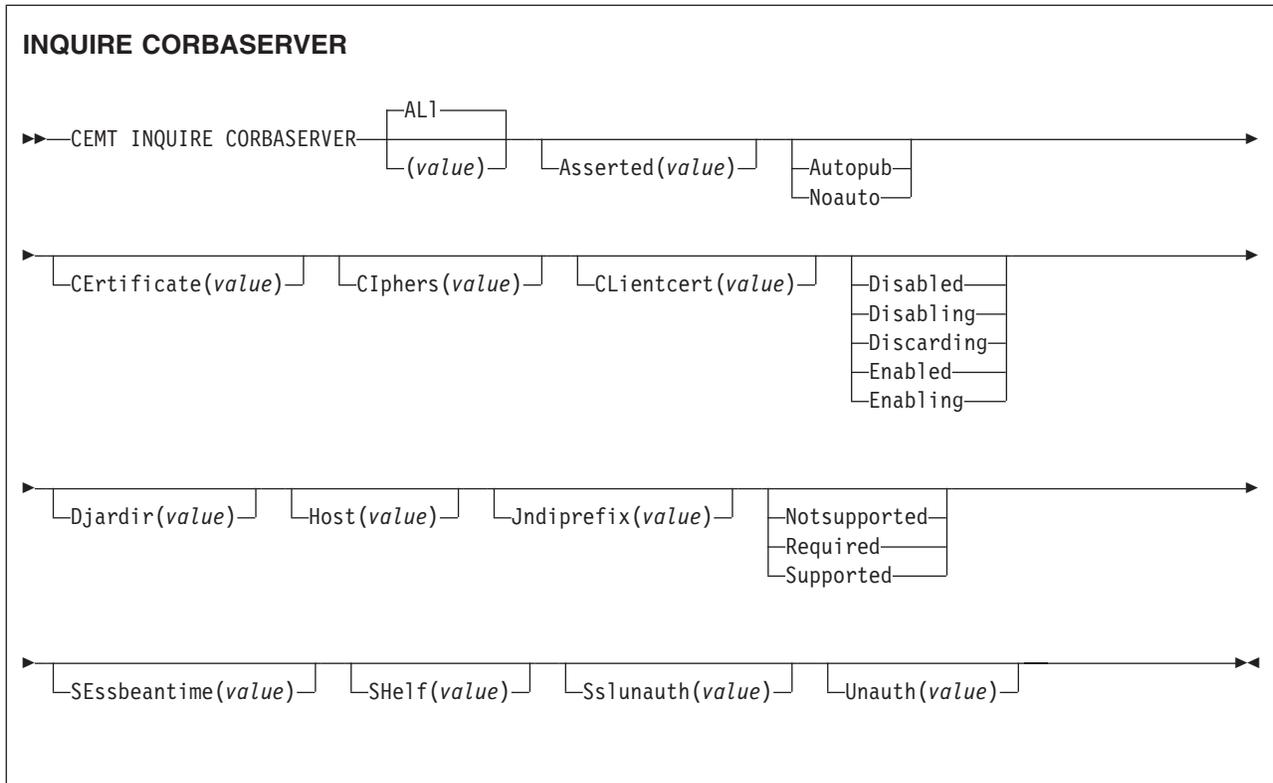
                                     SYSID=C2T1 APPLID=IYK4Z2T1
RESPONSE: NORMAL                                TIME: 11.19.46 DATE: 08.14.01
PF 1 HELP          3 END          5 VAR          7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 79. CEMT INQUIRE CORBASERVER screen

```
i corba
RESULT - OVERTYPE TO MODIFY
Corbaserver(CBS1)
Outprivacy(Supported)
Autopublish( Noauto )
Enablestatus(Enabled)
Sessbeantime( 000060 )
Unauth(IIOPUNA)
Clientcert()
Ssslunauth(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 80. CEMT INQUIRE CORBASERVER expanded display



**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.

**AL1**

is the default.

## Displayed fields

**ASserted***(value)*

returns the 8-character name of a TCPIP SERVICE that defines the characteristics of the port which is used for inbound IIOF 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 IIOF connections.

If this option is not specified, the default certificate for the key ring is used.

## CEMT INQUIRE CORBASERVER

### **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 IIOPI 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 HFS. (The pickup directory is the place that you put deployed JAR files that you want to be installed into the CorbaServer by the CICS scanning mechanism.)

### **Enablestatus**

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 HFS. 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 TCPIP SERVICE that defines the characteristics of the port which is used for inbound IIOPI with SSL and no authentication.

**Unauth**(*value*)

returns the 8-character name of a TCPIP SERVICE that defines the characteristics of the port which is used for inbound IIOPI with no authentication.

## CEMT INQUIRE DB2CONN

## 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.

Note that because there can be only one DB2CONN 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. You get a display that lists the current status. You can then tab to the highlighted or blank fields and overwrite them with the required values.

### Sample screen

```
inquire db2conn
STATUS: RESULTS - OVERTYPE TO MODIFY
Accountrec( Txid )           Planexitname( DSNCUEXT )
Authid(           )         Priority( High )
Authtype( Userid )         Purgecyclm( 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 81. CEMT INQUIRE DB2CONN screen

## CEMT INQUIRE DB2CONN

▶—CEMT Inquire DB2Conn—▶

### Displayed fields

#### Accountrec

returns the minimum amount of DB2 accounting required for transactions using pool threads. The specified minimum may be exceeded, as described in the following options. 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 to be produced for each CICS task.  
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.  
Because pool threads are typically used by a number of different transaction IDs, there is an increased chance that a transaction containing multiple UOWs will use a different thread for each UOW (assuming the thread is released at syncpoint. In this case, an accounting record may be produced for each UOW.
- 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 be used for security checking when using pool threads. If an AUTHID is returned, AUTHType is not applicable.

**Authtype**

returns the type of id to be used for security checking when using pool threads. If an AUTHType is returned, AUTHid is blank. 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 <sup>®</sup> 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.

- Sign** If no RACF group ID is available for this USERID, an 8-character field of blanks is passed to DB2 as the group ID. The SIGNID parameter of the DB2CONN is used as the resource authorization ID.
- Term** The terminal identification (four characters padded to eight) is used as an authorization ID. An authorization ID cannot be obtained in this manner if a terminal is not connected with the transaction.  
If a transaction is started (using a CICS command) and has no terminal associated with it, AUTHTYPE(TERM) should not be used.
- Tx** The transaction identification (four characters padded to eight) is used as the authorization ID.
- Opid** The user operator identification associated with the userid,

## CEMT INQUIRE DB2CONN

**Userid** associated with the CICS transaction, is used as the authorization ID (three characters padded to eight). The 8-character USERID associated with the CICS transaction is used as the authorization ID.

When the DB2 sample sign-on exit DSN3@SGN is used with AUTHTYPE(USERID), the exit sends the USERID to DB2 as the primary authorization ID and the RACF group ID to DB2 as the secondary ID. When the sample sign-on exit is used, there is no difference between AUTHTYPE(USERID) and AUTHTYPE(GROUP).

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

**Comauthtype** returns the type of ID to be used for security checking when using command threads. If COMAUTHType is returned, COMAUTHid is blank. The values are:

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

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.

**Csign** If no RACF group ID is available for this USERID, an 8-character field of blanks is passed to DB2 as the group ID. The SIGNID parameter of the DB2CONN is used as the resource authorization ID.

**Cterm** The terminal identification (four characters padded to eight) is used as an authorization ID. An authorization ID cannot be obtained in this manner if a terminal is not connected with the transaction.

If a transaction is started (using a CICS command) and has no terminal associated with it, COMAUTHTYPE(CTERM) should not be used.

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

**Copid** The operator identification associated with the userid that is associated with the CICS transaction is used as the authorization ID (three characters padded to eight).

**Cuserid** The 8-character USERID associated with the CICS transaction is used as the authorization ID.

When the DB2 sample sign-on exit DSN3@SGN is used with COMAUTHTYPE(CUSERID), the exit sends the USERID to DB2 as the primary authorization ID and the RACF group ID to DB2 as the secondary ID. When the sample sign-on exit is used, there is no difference between COMAUTHTYPE(CUSERID) and COMAUTHTYPE(CGROUP).

**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:

**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:

**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 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 subsystem CICS is connected to. When CICS is not connected to DB2, blanks are returned.

**Dro11back**

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:

**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.

## CEMT INQUIRE DB2CONN

### **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.

### **Nontermre1**

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

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

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

### **Plan**

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

### **Planexitname**

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

### **Priority**

returns the priority of the pool thread TCBs relative to the CICS main TCB (QR TCB). If CICS is connected to DB2 Version 6 or later, the thread TCBs are CICS open L8 TCBs. If CICS is connected to DB2 Version 5 or earlier, the thread TCBs are private TCBs created by the CICS-DB2 Attachment Facility. CVDA values are:

**HIGH** Thread TCBs have a higher priority than the CICS QR TCB.

**EQUAL** Thread TCBs have equal priority with the CICS QR TCB.

**LOW** Thread TCBs have a lower priority than the CICS QR TCB.

### **Purgecycl1em**

returns in minutes the length of the protected thread purge cycle. The range for PURGECYCLEM is 0-59.

A protected thread is not terminated immediately when it is released. It is terminated only after two completed purge cycles, if it has not been reused in the meantime. Hence if the purge cycle is set to 30 seconds after it is released, a protected thread will be purged 30 - 60 seconds after it is released. An unprotected thread is terminated when it is released (at syncpoint or end of task) if there are no other transactions waiting for a thread on that DB2ENTRY.

### **Purgecycles**

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

A protected thread is not terminated immediately when it is released. It is terminated only after two completed purge cycles, if it has not been reused in the meantime. Hence if the purge cycle is set to 30 seconds after it is released, a protected thread will be purged 30 - 60 seconds after it is released. An unprotected thread is terminated when it is released (at syncpoint or end of task) if there are no other transactions waiting for a thread on that DB2ENTRY.

**Resyncmember**

This applies only if you are using group attach (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 which are shunted indoubt are not included in this process, because CICS itself is unable to resolve those units of work at this time. Resynchronization for those UOWs will occur when CICS has resynchronized with its remote coordinator.) If you are not using group attach, a blank field is returned. CVDA values are:

**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**

has no meaning for INQUIRE. It is supplied to allow overtyping with the REBUILD option, to initiate a CEMT SET DB2CONN. See SECURITY on page "PURGECYCLES" on page 431.

**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:

**Noconnect** The CICS DB2 attachment terminates.  
**Connect** The CICS DB2 attachment goes into 'standby mode' to wait for DB2.  
**Reconnect** The CICS DB2 attachment goes into 'standby mode' and waits for DB2. Having connected to DB2, if DB2 subsequently fails the CICS DB2 attachment reverts again to standby mode and subsequently reconnects to DB2 when it comes up again.

**Statsqueue**

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

**Tcbs**

returns a number indicating the TCBs currently used by the CICS DB2 attachment facility. The number returned is the number of TCBs that are associated with DB2 connections (command, pool or DB2ENTRY threads), so the interpretation of the number depends on the release of DB2 to which CICS is connected, as follows:

**Connected to DB2 Version 5 or earlier (so not using the open transaction environment)**

Subtask TCBs are created and managed by the CICS DB2 attachment facility to service DB2 requests, and remain permanently associated with DB2 connections (command, pool or DB2ENTRY threads). In this case, the Tcbs option returns the high-water mark of TCBs created to access DB2.

**Connected to DB2 Version 6 or later (so using the open transaction environment)**

The TCBs used by the CICS DB2 attachment facility are allocated by CICS

## CEMT INQUIRE DB2CONN

from the pool of L8 mode TCBs. A DB2 connection is not permanently assigned to the same L8 TCB, and between CICS tasks, it can move from one L8 mode TCB to another. In this environment, the Tcbs option returns the number of L8 mode TCBs that are using a DB2 connection at the time of the inquiry, and this value varies depending on workload.

### **Tcblimit**

returns the maximum number of TCBs that can be used to process DB2 requests. When connected to DB2 Version 5 or earlier, the CICS DB2 attachment facility creates the TCBs in the form of subtasks up to the limit specified by TCBLIMIT. Each of these subtasks identifies to DB2 and creates a connection into DB2. When connected to DB2 Version 6 or later, CICS creates 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:

<b>Abend</b>	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.
<b>N906d</b>	A transaction dump is taken and the DSNCSQL RMI associated with the transaction is <i>not</i> 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.
<b>N906</b>	The DSNCSQL RMI associated with the transaction is <i>not</i> 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;

<b>Twait</b>	If all threads are busy, a transaction waits until one becomes available.
<b>Notwait</b>	If all threads are busy, a transaction is terminated with an abend code AD3T.

## CEMT INQUIRE DB2ENTRY

### Function

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)
      Thread1( 0000 ) Threads(0000) Tpo                Plane( DSNCUEXT
Db2e(DB2ENT2 ) Use Ena Poo Hig Pro( 0000 ) Pth(0000)
      Thread1( 0000 ) Threads(0000) Tpo                Plane( DSNCUEXT

```

Figure 82. CEMT INQUIRE DB2ENTRY screen

If you place the cursor against the result and press ENTER, an expanded format is displayed as shown in Figure 83.

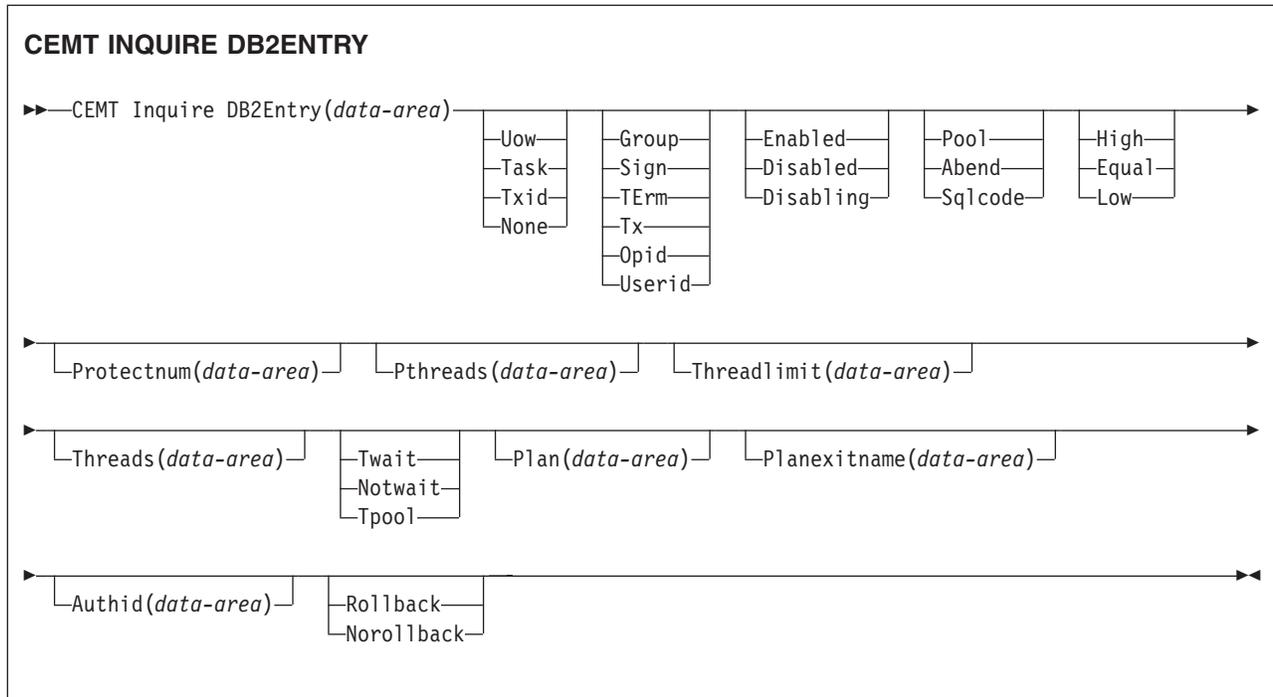
```

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 83. 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.

- Sign** 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. 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.
  - Tx** If a transaction is started (using a CICS command) and has no terminal associated with it, AUTHTYPE(TERM) should not be used. 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 MSGQUEUE on the DB2CONN for each transaction routed to the pool.
- Abend** The CICS DB2 attachment facility abends the transaction. The abend code is AD26.

## CEMT INQUIRE DB2ENTRY

**Sqlcode** An SQLCODE is returned to the application indicating that the DB2ENTRY is disabled.

### **Dro11back**

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). If CICS is connected to DB2 Version 6 or later, the thread TCBs are CICS open L8 TCBs. If CICS is connected to DB2 Version 5 or earlier, the thread TCBs are private TCBs created by the CICS-DB2 Attachment Facility. CVDA values are:

**HIGH** Thread TCBs have a higher priority than the CICS QR TCB.

**EQUAL** Thread TCBs have equal priority with the CICS QR TCB.

**LOW** Thread TCBs have a lower priority than the CICS QR TCB.

### **Protectnum**

returns the maximum number of protected threads allowed for this DB2ENTRY.

### **Pthreads**

returns the current number of protected threads for this DB2ENTRY. A protected thread is an inactive thread available for reuse by a new transaction. If no transaction has reused the thread by the time it has been processed by 2 purge cycles, the thread is terminated.

### **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

### Function

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 DFHttt, where tttt is the TRANSID.

### Sample screen

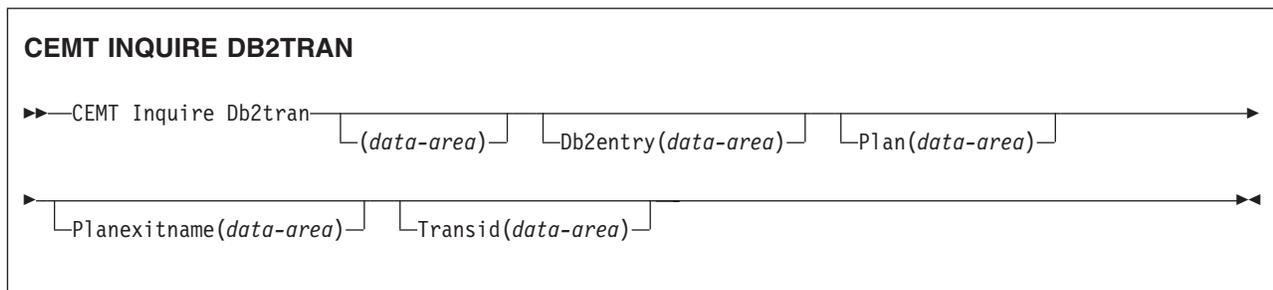
```
inquire db2tran
STATUS: RESULTS - OVERTYPE TO MODIFY
Db2t(DFHTXN* ) Db2e( DB2ENT2 ) Tra( TXN* ) Plane(DSNCUEXT)
Db2t(DFHTXN2 ) Db2e( DB2ENT1 ) Tra( TXN2 ) Plane(DSNCUEXT)
Db2t(TXN1 ) Db2e( DB2ENT1 ) Tra( TXN1 ) Plane(DSNCUEXT)
```

Figure 84. CEMT INQUIRE DB2TRAN screen

If you place the cursor against the result and press ENTER, an expanded format is displayed as shown in Figure 85.

```
inquire db2tran
RESULT - OVERTYPE TO MODIFY
Db2tran(DFHTXN*)
Db2entry( DB2ENT2 )
Transid( TXN* )
Plan()
Planexitname(DSNCUEXT)
```

Figure 85. The expanded display of an individual entry



### 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* for information about use of wildcard characters).

# CEMT INQUIRE DELETSHIPED

### Function

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 DELETSHIPED displays the current settings of the parameters that control the mechanism.

### Input

Press the Clear key and type CEMT INQUIRE DELETSHIPED (the minimum abbreviation is CEMT I DE). You will get a display screen.

To change attributes, you can:

- Overtyping your changes on the INQUIRE screen after tabbing to the appropriate field.
- Use the CEMT SET DELETSHIPED command.

### Sample screen

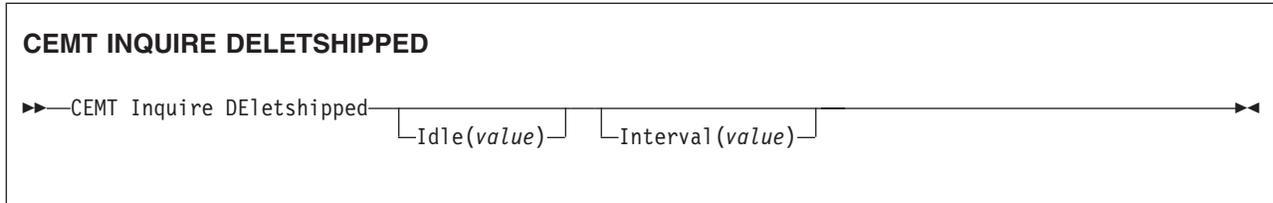
```
IN DELE
STATUS: RESULTS - OVERTYPE TO MODIFY
De1 Int( 120000 ) Id1( 020000 )
```

Figure 86. CEMT INQUIRE DELETSHIPED screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 87.

```
IN DELE
RESULT - OVERTYPE TO MODIFY
Deletshipped
Interval( 120000 )
Idle( 020000 )
```

Figure 87. The expanded display



## Displayed fields

### De1

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

## CEMT INQUIRE DISPATCHER

### Function

Retrieve system information used by the CICS dispatcher.

# **Note:** Except for ACTxxxxTCBS and MAXxxxxTCBS, you can also inquire on these dispatcher system values using the CEMT INQUIRE SYSTEM command.

### 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:

- Overtyping 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)
Actsslctcbs(000)
Actxptcbs(000)
Aging( 32768 )
Maxjvmtcbs( 005 )
Maxopentcbs( 005 )
Maxsslctcbs(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 88. CEMT INQUIRE DISPATCHER screen

## CEMT INQUIRE DISPATCHER

▶▶—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).

|           **ACTSs1tcbs**(*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.

|           **MAXSs1tcbs**(*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.

## CEMT INQUIRE DISPATCHER

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.

#### **Notes:**

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'.

#### **Notes:**

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

### Function

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) Hfsfi(/u/ivanh/ivanejbd.jar  )
Djar(IJAR                               ) Corba(CORA) Inser
  Dates(20010222) Times(14:33:52) Hfsfi(/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 89. CEMT INQUIRE DJAR screen

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format.

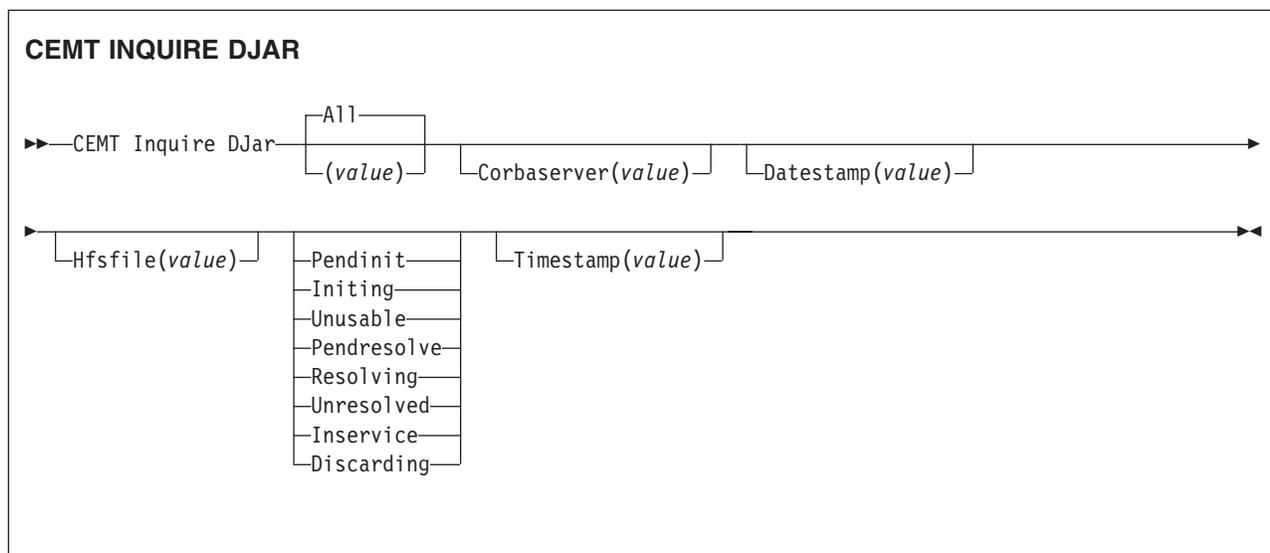
```

inq djar
RESULT
Djar(BJAR)
Corbaserver(CORB)
State(Inservice)
Datestamp(20010923)
Timestamp(17:39:48)
Hfsfile(/u/ivanh/ivanejbd.jar)

```

Figure 90. The expanded display of an individual entry

## CEMT INQUIRE DJAR



**(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 HFS. 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 HFS file pointed to by the DJAR definition, at the time the DJAR definition was last installed or updated.

For statically-installed DJARs (those installed from a CSD or by CREATE DJAR), the value of DATESTAMP is the date of the HFS 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.

**Hfsfile(value)**

displays a 255-character, fully-qualified HFS file name.

**State(value)**

The values are:

- |                 |   |
|-----------------|---|
| <b>Pendinit</b> | The DJar has just been created and is awaiting Stage 1 Resolution.  |
| <b>Initing</b>  | 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. |

<b>Unusable</b>	The DJar has failed Stage 1 Resolution and is not available for use.
<b>Pendresolve</b>	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).
<b>Resolving</b>	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.
<b>Unresolved</b>	The DJar has failed Stage 2 Resolution, or some other processing, and is not available for use; Beans were not correctly loaded).
<b>Inservice</b>	The DJar has undergone both Stage 1 and Stage 2 Resolution and is available for use; its Beans are all correctly loaded as well.
<b>Discarding</b>	The DJar is being deleted.

**Timestamp**(*value*)

displays the time, in hh:mm:ss format, of the deployed JAR file on HFS. 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 HFS file pointed to by the DJAR definition, at the time the DJAR definition was last installed or updated.

For statically-installed DJARs (those installed from a CSD or by CREATE DJAR), the value of TIMESTAMP is the time of the HFS 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

### Function

Retrieves information about a DOCTEMPLATE

### Description

INQUIRE DOCTEMPLATE returns information about any currently installed document template names.

### Input

Press the Clear key to clear the screen, and type CEMT INQUIRE DOCTEMPLATE (the minimum abbreviation is CEMT I D0). 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

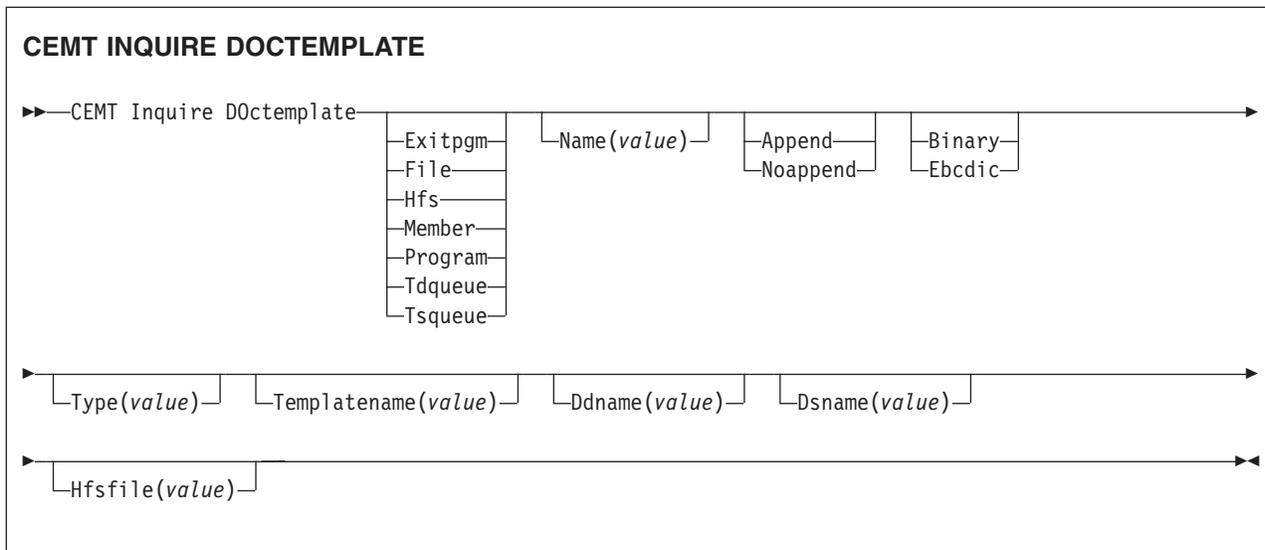
```
I DOC
STATUS: RESULTS - OVERTYPE TO MODIFY
Doc(test1  ) Tsq Nam(test1      ) App Ebc
    Tem(test1                      )
Doc(FORMAPI2) Mem Nam(FORMAPI2  ) App Ebc
    Tem(FORMAPI2                    ) Ddn(WEBTESTD)
Doc(SKINNER ) Hfs Nam(index.html) App Ebc
    Tem(SKINNER                      )
```

Figure 91. 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(SKINNER)
Templatetype(Hfs)
Name(index.html)
AppendCrLf(Append)
Type(Ebcdic)
Templatename(SKINNER)
Ddname()
Dsname()
Hfsfile(/u/coultha/website/index.html)
```

Figure 92. The expanded display of an individual entry



## 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 DDName applies only to a template of type PDS. If a membername is supplied without a value for DDNAME, the default value DFHTML is used.

### DOctemplate

returns the name of this DOCTEMPLATE definition.

### DSname

returns the DS name of the PDS containing the template.

### Hfsfile

returns the fully-qualified name of the z/OS UNIX System Services HFS 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.

### Templatename

returns the extended template name by which the doctemplate is known outside the resource definition function.

### 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 System Services HFS file.

**Member** a name of the member in the PDS described in DDNAME.

## CEMT INQUIRE DOCTEMPLATE

**Program** a name of a program.

**Tdqueue** a name of a TD queue.

**Tsqueue** a name of 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

### Function

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:

- Overtyping your changes on the INQUIRE screen after tabbing to the appropriate field (see Figure 93)
- Use the CEMT SET DSAS command.

### Sample screen

```

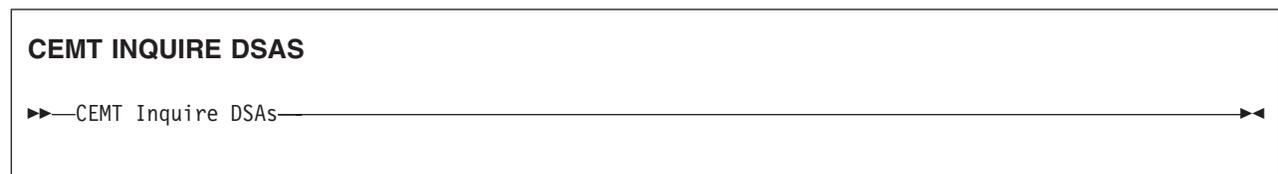
IN DSAS
STATUS: RESULTS - OVERTYPE TO MODIFY
      Sosstatus(Notsos)

      Dsalimit( 04194304 )
      Cdsasize( 00524288 )
      Rdsasize( 00262144 )
      SDsasize( 00262144 )
      Udsasize( 00000000 )

      EDsalimit( 0020971520 )
      ECdsasize( 0002097152 )
      ERdsasize( 0003145728 )
      ESdsasize( 0000000000 )
      EUdsasize( 0001048576 )

```

Figure 93. CEMT INQUIRE DSAS screen



### 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.)

## CEMT INQUIRE DSAS

**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.

### **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.

### **Sosstatus**

displays whether CICS is short on storage in any of the dynamic storage areas. The values are:

**Notsos** CICS is not short on storage in any of the dynamic storage areas.

**Sos** CICS is short on storage in dynamic storage areas above and below 16MB.

**Sosabove** CICS is short on storage in at least one of the dynamic storage areas above 16MB.

**Sosbelow** CICS is short on storage in at least one of the dynamic storage areas below 16MB.

**Udsasize**(*value*)

displays the size, in bytes, of the user dynamic storage area (UDSA) below the 16MB boundary.

## CEMT INQUIRE DSNNAME

### Function

Retrieve information about an external data set.

### Description

INQUIRE DSNNAME 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 DSNNAME 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 DSNNAME (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 DSNNAME (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 overwrite them with the required values.

**AL1**

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(CICSTS31.CICS.CICSH###.DFHCSD          ) Vsa
   Fil(0001) Val Bas      Sta              Ava
Dsn(CICSTS31.CICS.CICSHT61.FILEA          ) Vsa
   Fil(0001) Val Bas      Sta              Ava
```

Figure 94. CEMT INQUIRE DSNNAME 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:

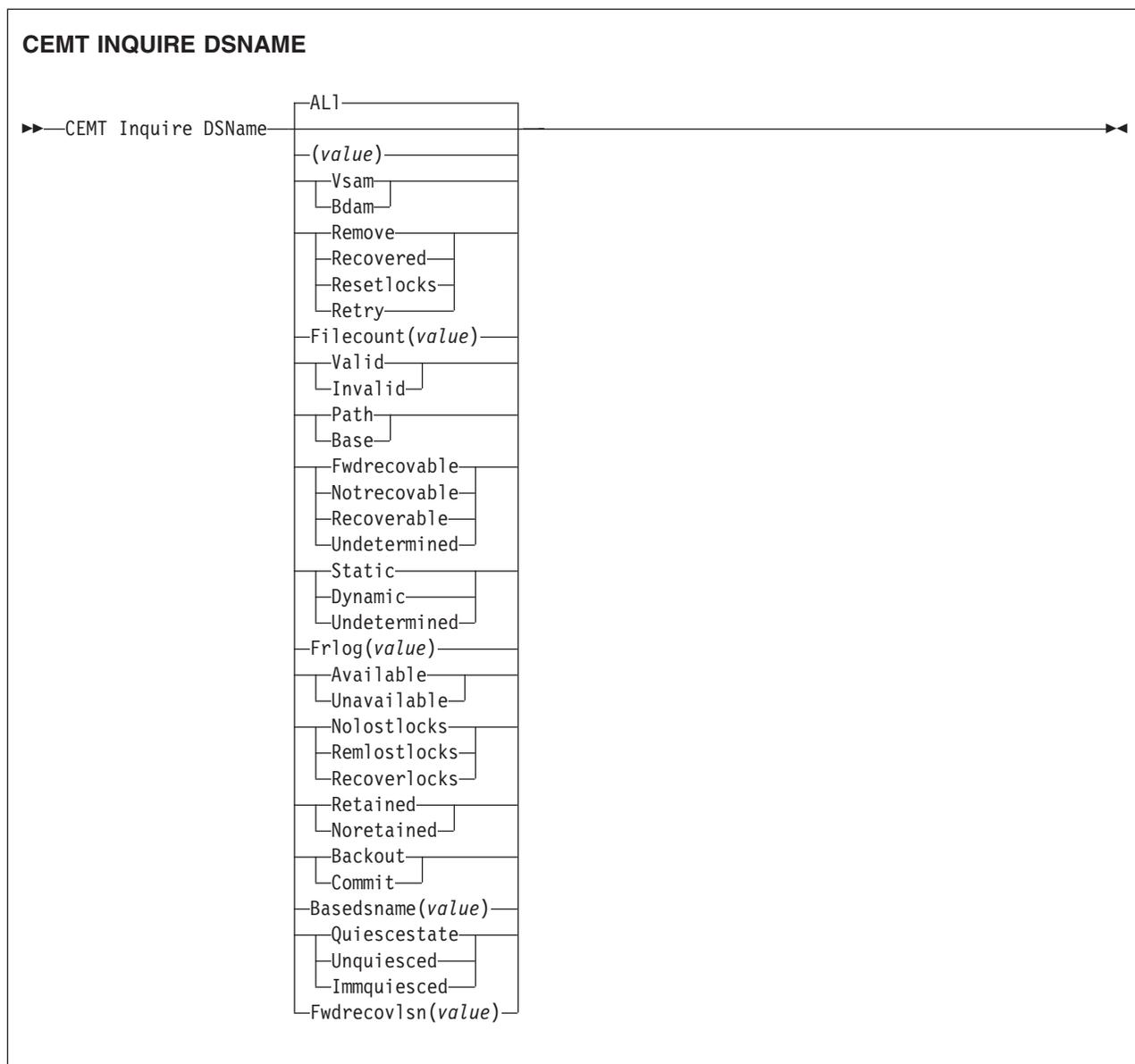
```

IN DSN
RESULT - OVERTYPE TO MODIFY
  Dsname(CICSTS31.CICS.CICSH###.DFHCSD)
  Accessmethod(Vsam)
  Action(          )
  Filecount(0001)
  Validity(Valid)
  Object(Base)
  Recovstatus(Notrecoverable)
  Backuptype(Static)
  Frlog()
  Availability( Available )
  Lostlocks()
  Retlocks(Noretained)
  Uowaction(          )
  Basedsname(CICSTS31.CICS.CICSH###.DFHCSD)
  Quiescestate( Unquiesced )
  Fwdrecovlsn()

```

Figure 95. 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.

### Notes:

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<sup>®</sup> 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™.<sup>2</sup>

#### 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.

#### Fwdrecovl sn (*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

2. 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.

## CEMT INQUIRE DSNAME

- 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:

**Fwdrecoverable**

All updates to the data set are logged for both backout and forward recovery.

**Notrecoverable**

Updates to the data set are not logged.

This response may also be returned as the result of use of the XFCNREC global user exit. A program enabled at XFCNREC may indicate that file opens should proceed even if there is a mismatch in the backout recovery requirements for different files associated with same data set. In these circumstances, the data set is marked as NOTRECOVERABLE to indicate that its data integrity can no longer be guaranteed.

The condition remains until cleared by a CEMT 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

## CEMT INQUIRE DSNAME

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

### Function

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 overwrite your changes on the INQUIRE screen after tabbing to the appropriate field. See “Overtyping a display” on page 140.

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 overwrite them as required.

### Sample screen

```
IN DUMPDS
STATUS: RESULTS - OVERTYPE TO MODIFY
Dum Cur(A) Ope
```

Figure 96. CEMT INQUIRE DUMPDS screen

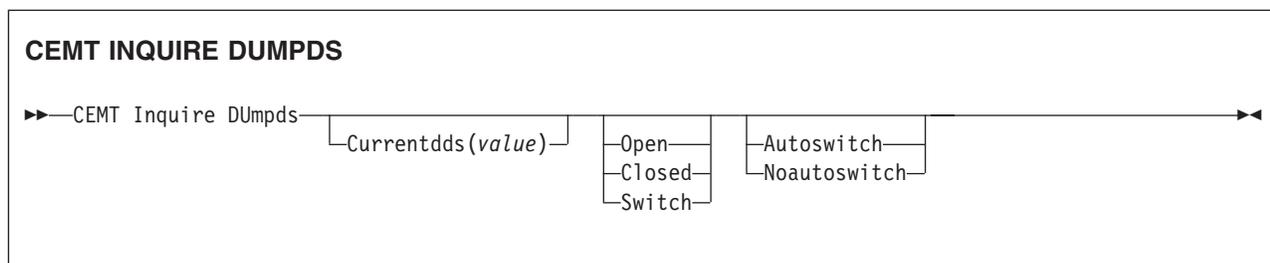
**Note:** There are blank fields on the screen where a value does not apply or is ‘negative’; that is, the value begins with ‘No’. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overwrite with input valid for that field. You may find it 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 97.

```
IN DUMPD
RESULT - OVERTYPE TO MODIFY
Dumpds
Currentdds(A)
Openstatus( Open )
Switchstatus( Noautoswitch )
```

Figure 97. The expanded display

## CEMT INQUIRE DUMPDS



### Displayed fields

#### Currentdds (*value*)

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

#### Dumps

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

#### Openstatus

displays 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

displays 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

### Function

–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.

### Notes:

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 in the *CICS System Programming Reference*).
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 98. CEMT INQUIRE ENQ screen

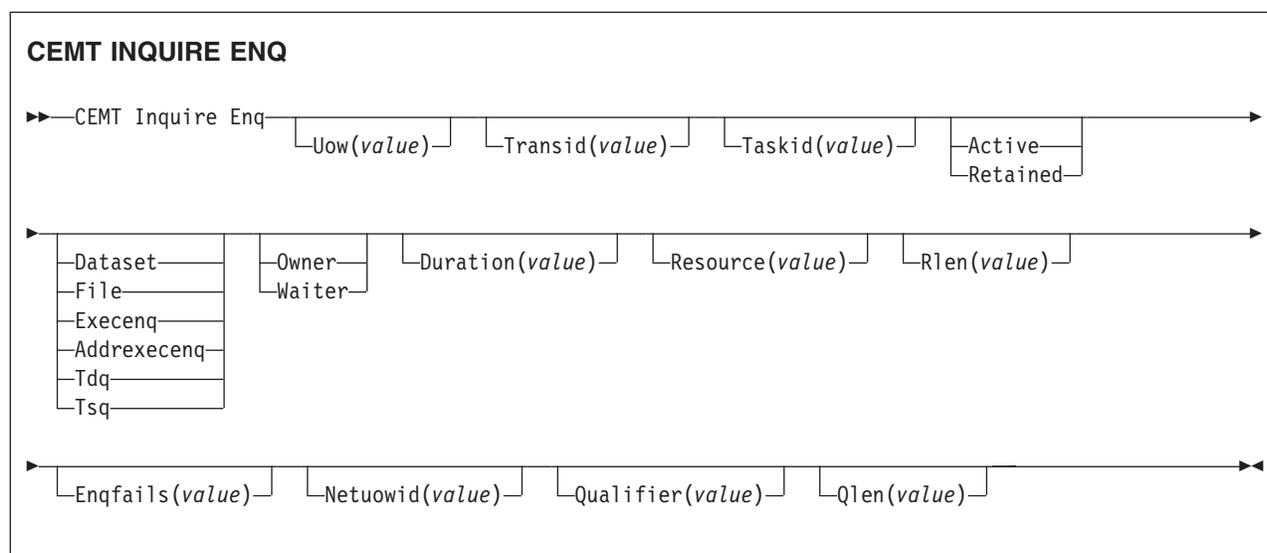
**Note:** There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overwrite with input valid for that field. You may find it 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 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 99. The expanded display of an individual entry



## 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.

### 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 229. 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.

## CEMT INQUIRE ENQ

### 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 229.

**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.
- Addrxecenq** 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

**Function**

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 overwrite them as required.

**Sample screen**

```

INQ ENQMODEL
STATUS: RESULTS - OVERTYPE TO MODIFY
Enqm(*          )                NOT FOUND

```

Figure 100. 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          )

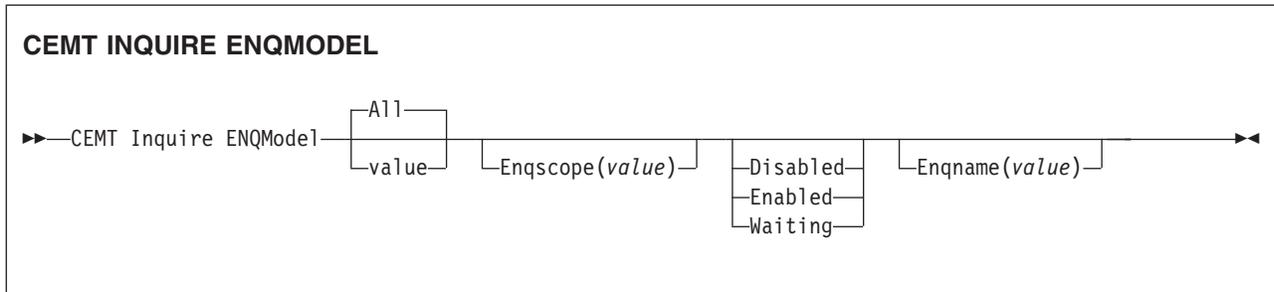
                                SYSID=HT61 APPLID=CICSHT61
RESPONSE: NORMAL                TIME: 13.23.25 DATE: 02.21.01
PF 1 HELP          3 END          5 VAR          7 SBH 8 SFH 9 MSG 10 SB 11 SF

```

Figure 101. 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 overwrite 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).



### Displayed fields

**A11**

is sthe 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

### Function

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 102. CEMT INQUIRE EXCI screen

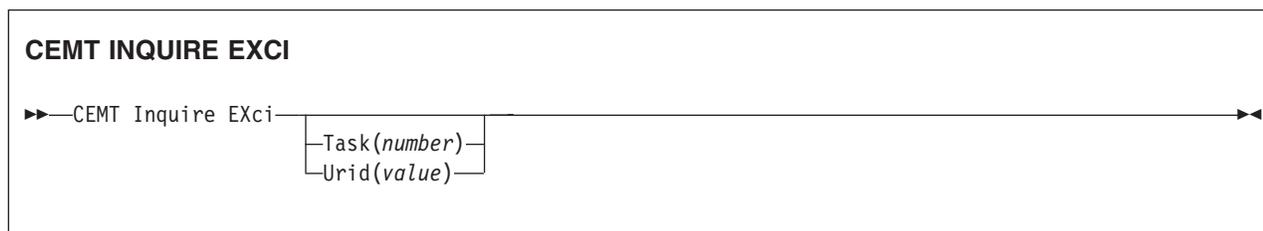
**Note:** There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overwrite with input valid for that field. You may find it 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 103.

```

INQ EXCI
RESULT
Exc(HADBS1..CICSWK1 - MV29           ) Tas(0000121)
Task(0000121)
    
```

Figure 103. The expanded display of an individual entry



**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

## 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

### Function

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 overwrite them with the required values.

(*value*)

specifies one or more names (1-8 characters) of files defined to CICS.

AL1

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( C1CSTS31.C1CS.DFHCMACD
           )
Fil(DFHCS D ) Vsa Clo Une Rea Upd Add Bro Del  Sha
           Dsn( C1CSTS31.C1CS.C1CSH.DFHCS D
           )
Fil(DFHDBFK ) Vsa Clo Ena Rea Upd Add Bro Del  Sha
Fil(DFH LRQ ) Vsa Ope Ena Rea Upd Add Bro Del  Sha
           Dsn( C1CSTS31.C1CS.C1CSH.DFH LRQ
           )
Fil(FILEA ) Vsa Clo Ena Rea Upd Add Bro Del  Sha
           Dsn( C1CSTS31.C1CS.C1CSH.FILEA
           ) Cft Loa

```

Figure 104. CEMT INQUIRE FILE screen

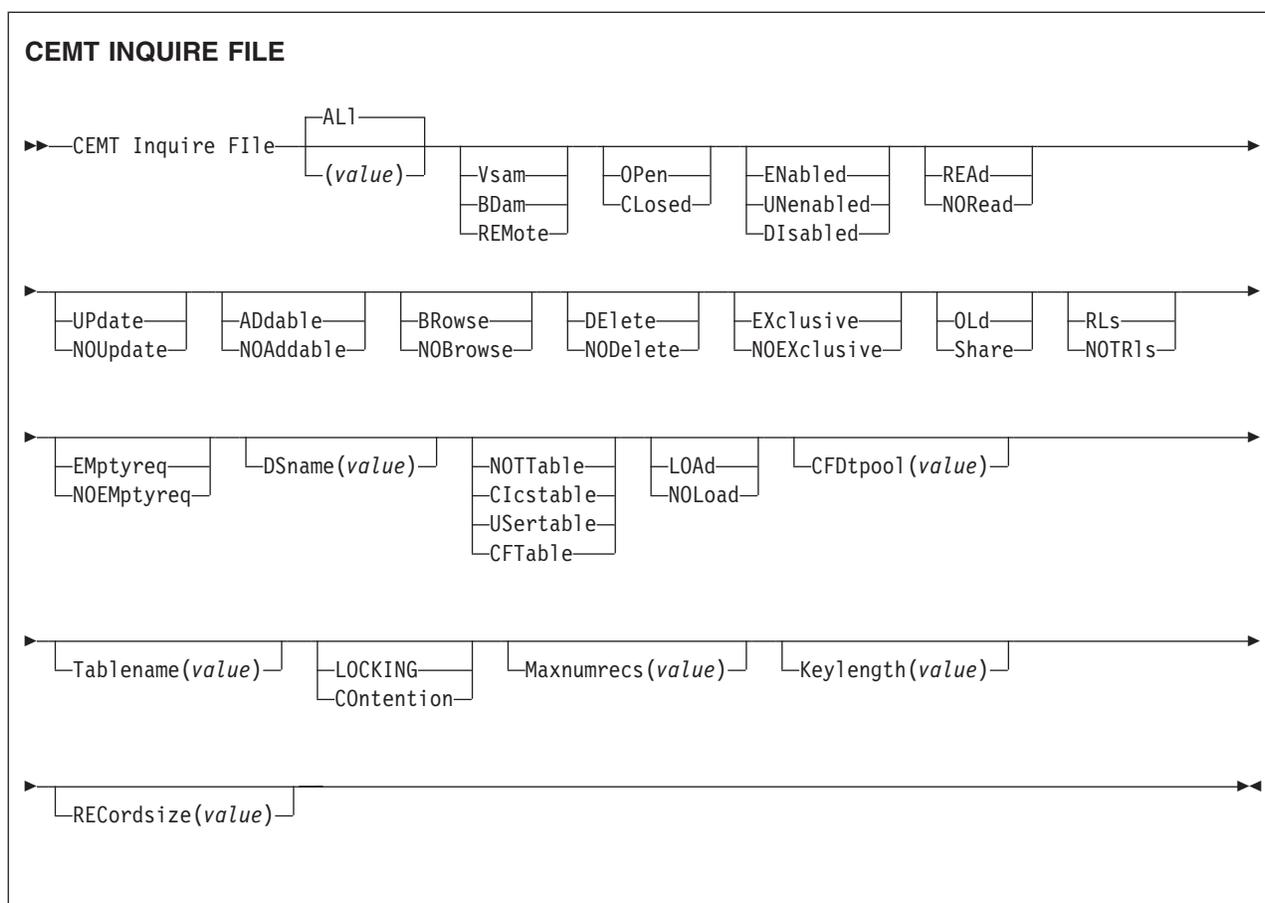
**Note:** There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overwrite with input valid for that field. You may find it 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 FILE

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 105.

```
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( CICSTS31.CICS.CICSH.DFHCSD          )
Table( Nottable )
Loadtype( Noload )
Cfdtpool(          )
+ Tablename(          )
  Updatemodel(          )
  Maxnumrecs( 00000000 )
  Keylength( 000 )
  Recordsize( 00000 )
```

Figure 105. The expanded display of an individual entry



## 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.

## CEMT INQUIRE FILE

### **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.

**Notes:**

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.

## CEMT INQUIRE FILE

### Notes:

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.

### 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:

<b>Rls</b>	The file is defined to be accessed in RLS mode.
<b>Notrls</b>	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:

<b>Nottable</b>	The file name does not represent a data table.
<b>Cftable</b>	The file name represents a coupling facility data table.
<b>Cicstable</b>	The file name represents a CICS-maintained data table.
<b>Usertable</b>	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

**Updatemode1** (*CFDT only*)

displays the update model specified for the coupling facility data table in the installed file definition. The values are:

<b>Contention</b>	The coupling facility data table is updated using the contention model.
<b>Locking</b>	The coupling facility data table is updated using the locking model.

**Updatestatus**

displays whether the file is updatable. The values are:

<b>Update</b>	You can update records in the file.
<b>Noupdate</b>	You cannot update records in the file.

**Note:** You can reset this value by overtyping it with a different value.

## CEMT INQUIRE GTFTRACE

### Function

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 overwrite it as required.

### Sample screen

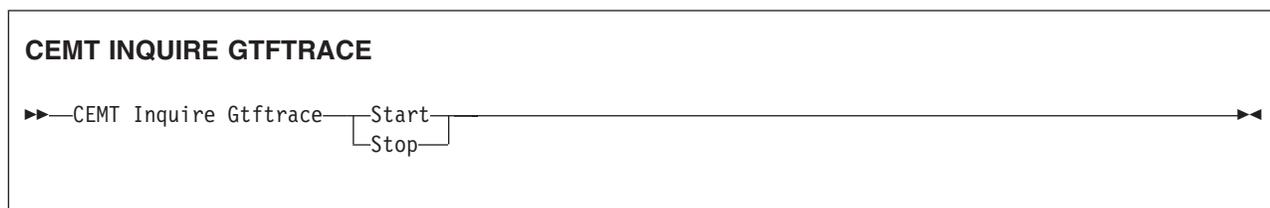
```
IN GTFTRA
STATUS: RESULTS - OVERTYPE TO MODIFY
Gtf Sto
```

Figure 106. CEMT INQUIRE GTFTRACE screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 107.

```
IN GTFT
RESULT - OVERTYPE TO MODIFY
Gtftrace
Status( Stop )
```

Figure 107. The expanded display



**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

### CEMT INQUIRE HOST

#### Function

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 TCPIP SERVICE. URIMAP definitions that specify no TCPIP SERVICE, 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 overwrite 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 108. CEMT INQUIRE HOST screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 109.

```
IN HOST
RESULT - OVERTYPE TO MODIFY
Host(www.hursleytest.ibm.com)
Tcpiptime( TEST2 )
Enablestatus( Enabled )
```

Figure 109. The expanded display

### CEMT INQUIRE HOST

```
▶ CEMT Inquire Host (value) Tcpiptime(value) Enabled
                   AL1         Disabled
```

#### 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.research.ibm.com, CICS would create a virtual  
| host with the name www.research.ibm.com. A host name in a URIMAP definition  
| can be up to 120 characters.

| **Tcpipservice** (*value*)

| displays the name of the TCPIP SERVICE definition that specifies the inbound  
| port to which this virtual host relates. If this definition is not given, the virtual  
| host relates to all TCPIP SERVICE 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

## CEMT INQUIRE INTTRACE

### Function

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 overwrite it as required.

### Sample screen

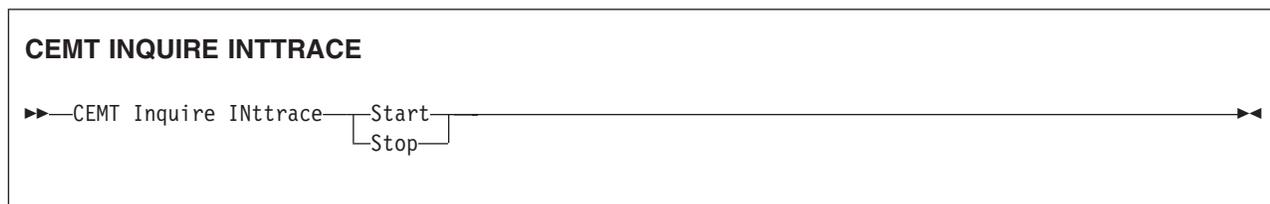
```
IN INTTRA
STATUS: RESULTS - OVERTYPE TO MODIFY
Int Sta
```

Figure 110. CEMT INQUIRE INTTRACE screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 111.

```
IN INTT
RESULT - OVERTYPE TO MODIFY
Intrace
Status( Start )
```

Figure 111. The expanded display



### Displayed fields

#### Intrace

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**  
**Stop**

CICS internal tracing is in progress.  
CICS internal tracing is not in progress.

**Note:** You can reset this value by overtyping it with a different value.

## CEMT INQUIRE IRBATCH

## CEMT INQUIRE IRBATCH

### **Function**

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

### Function

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

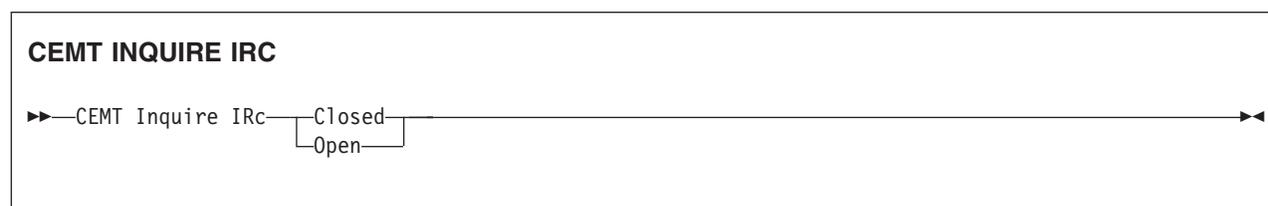
```
IN IRC
STATUS: RESULTS - OVERTYPE TO MODIFY
Irc Cl
```

Figure 112. CEMT INQUIRE IRC screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 113.

```
IN IRC
RESULT - OVERTYPE TO MODIFY
Irc
Openstatus( Closed )
Return(00000000)
Reason(00000000)
```

Figure 113. The expanded display



### 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:

## CEMT INQUIRE IRC

- 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.

### Notes:

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.

## CEMT INQUIRE JMODEL

### Function

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.

A11

is the default.

### Sample screen

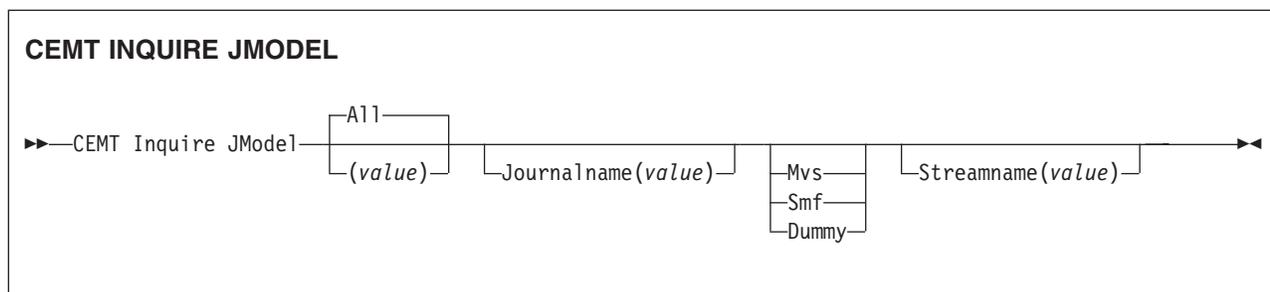
```
IN JMODE
STATUS: RESULTS
Jmo(DFHJ02 ) Jou(DFHJ02 ) Mvs
  Str(BELL.TESTLOG.FWDRECOV )
Jmo(DFHLOG ) Jou(DFHLOG ) Mvs
  Str(&USERID..&APPLID..&JNAME. )
Jmo(DFHSHUNT) Jou(DFHSHUNT) Mvs
  Str(&USERID..&APPLID..&JNAME. )
```

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 114.

```
IN JMODE
RESULT
Jmodel (DFHJ02)
Journalname(DFHJ02)
Type(Mvs)
Streamname(BELL.TESTLOG.FWDRECOV)
```

Figure 114. 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:

- |              |   |
|--------------|---|
| <b>Mvs</b>   | Records for journals defined with this journal model are written to an MVS log stream.    |
| <b>Smf</b>   | Records for journals defined with this journal model are written to the MVS SMF data set. |
| <b>Dummy</b> | Records for journals defined with this journal model are not written to any log stream.   |

For information on changing JMODEL definitions, see *CICS Resource Definition Guide*.

## CEMT INQUIRE JOURNALNAME

### Function

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 J0). You get a display that lists the current status.
- Type CEMT INQUIRE JOURNALNAME (CEMT I J0) 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 `cent 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 `DFHJnn`, where *nn* is the journal number.

To inquire on the system log, specify `DFHLOG`.

To inquire on the log-of-logs, specify `DFHLGLOG`.

A11

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 115. CEMT INQUIRE JOURNALNAME screen

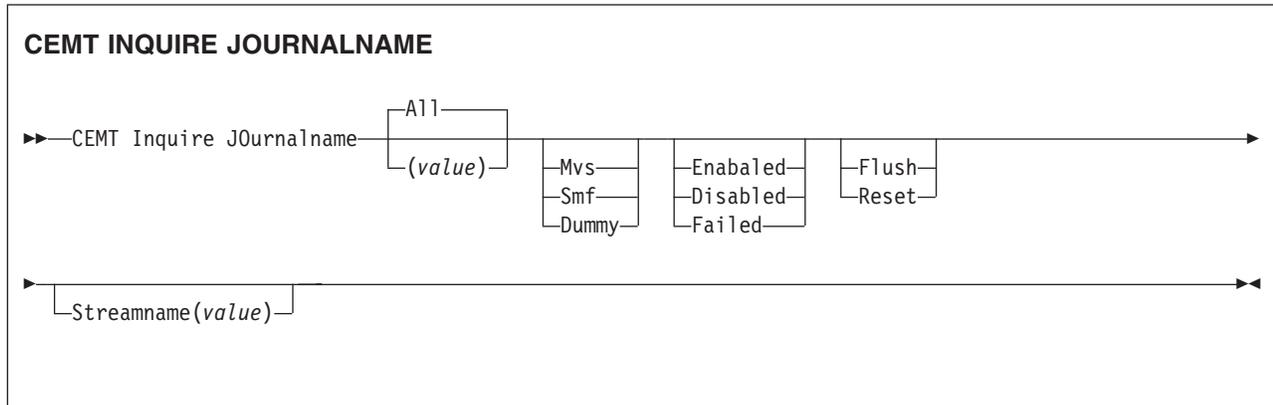
**Note:** Blank fields on the screen indicate that a value does not apply or is 'negative'; that is, it begins with 'No'.

If you place the cursor against a specific entry in the list, CICS displays an expanded format as shown in Figure 116 on page 254.

## CEMT INQUIRE JOURNALNAME

```
IN JOUR
RESULT - OVERTYPE TO MODIFY
Journalname(DFHLOG)
Type(Smf)
Status( Enabled )
Action(          )
Streamname(CICSHT##.CICSHTA1.DFHLOG)
```

Figure 116. The expanded display



### 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 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:

- |              |   |
|--------------|---|
| <b>Mvs</b>   | Records for journals defined with this journal model are written to an MVS log stream.    |
| <b>Smf</b>   | Records for journals defined with this journal model are written to the MVS SMF data set. |
| <b>Dummy</b> | Records for journals defined with this journal model are not written to any log stream.   |

## CEMT INQUIRE JVM

### Function

Identify the 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.

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.

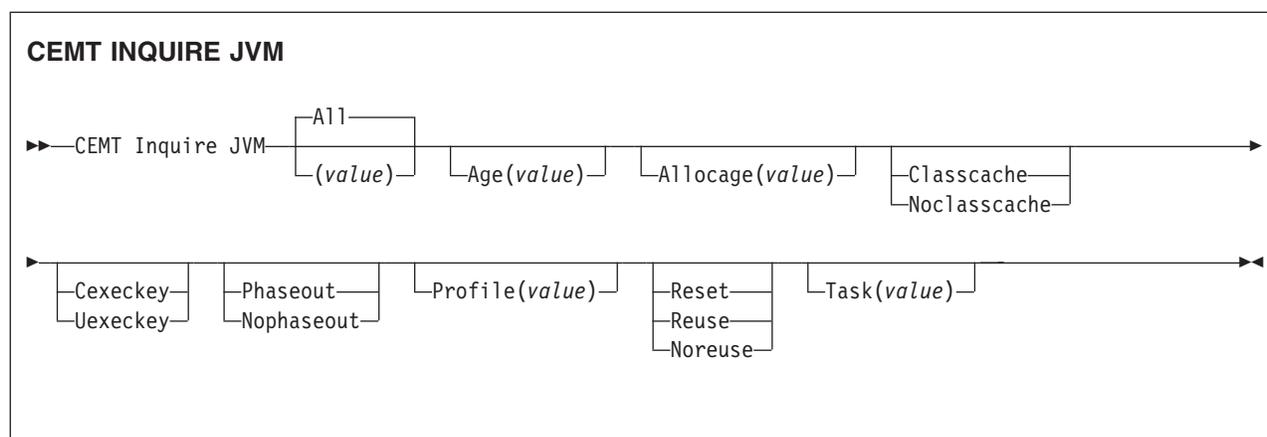
### Sample screen

```
i jvm
STATUS: RESULTS - OVERTYPE TO MODIFY
Jvm(0050463184) Age(0000001810) Allo(0000000000) Cla Res
  Uex    Pro(DFHJVMPC) Tas(0000000)
Jvm(0050463485) Age(0000001804) Allo(0000000000)      Reu
  Uex    Pro(DFHJMPCR) Tas(0000000)
Jvm(0067240142) Age(0000001711) Allo(0000000000)      Reu
  Uex    Pro(DFHJMPCR) Tas(0000000)
Jvm(0084017268) Age(0000001668) Allo(0000000000) Cla Res
  Uex    Pro(DFHJVMPC) Tas(0000000)
```

Figure 117. CEMT INQUIRE JVM screen

```
INQ JVM
RESULT - OVERTYPE TO MODIFY
  Jvm(0050463184)
  Age(0000001811)
  Allocage(0000000000)
  Classcachest(Classscache)
  Reusest(Reset)
  Execkey(Uexeckey)
  Phasingoutst(Nophaseout)
  Profile(DFHJVMPC)
  Task(0000000)
```

Figure 118. The expanded display



## Displayed fields

### *(value)*

# is the JVM token, a numeric value that identifies the JVM. The JVM token used  
# in CICS is the same number that is used to identify the JVM in the z/OS UNIX  
# environment, where it is known as the process id (PID) for the JVM.

### **Age***(value)*

displays the number of seconds since the JVM was initialized.

### **Allcage***(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 is a worker JVM dependent on 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.

## CEMT INQUIRE JVM

### **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 the level of reusability for this JVM. The values are:

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

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

### **Noreuse**

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

### **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

### Function

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.

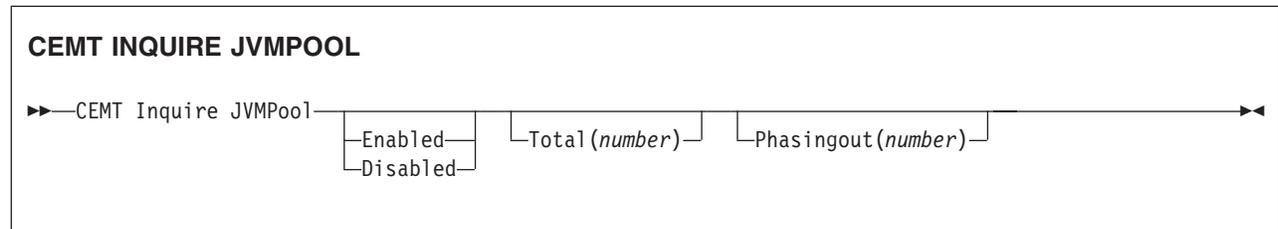
### Sample screen

```
i jvmp
STATUS: RESULTS - OVERTYPE TO MODIFY
Jvm Ena Tot(0004) Phasi(0000)
```

Figure 119. CEMT INQUIRE JVMPOOL screen

```
INQ JVMPOOL
RESULT - OVERTYPE TO MODIFY
Jvmpool
Status( Enabled )
Total(0004)
Phasingout(0000)
Terminate( )
```

Figure 120. The expanded display



### 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.

JVMs are marked for removal 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).

#### 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

## CEMT INQUIRE JVMPOOL

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 LINE

### Function

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 of commencing this transaction:

- Type CEMT INQUIRE LINE(*value*) (the minimum abbreviation is CEMT I L(*value*)). You get a display that lists the current status of the named line.
- Type CEMT INQUIRE LINE(*termid*) (CEMT I L(*termid*)) followed by as many of the other attributes as are necessary to limit the range of information that you require.

You can specify a list of identifiers (see “List of resource identifiers” on page 140), but you cannot use the symbols \* and + to specify a family of identifiers. You can then tab to the highlighted or blank fields and overwrite them with the required values.

### Sample screen

```
INQ LINE(TTT1)
STATUS: RESULTS
Lin(TTT1)      Neg( 00080 ) Ins
```

Figure 121. CEMT INQUIRE LINE screen

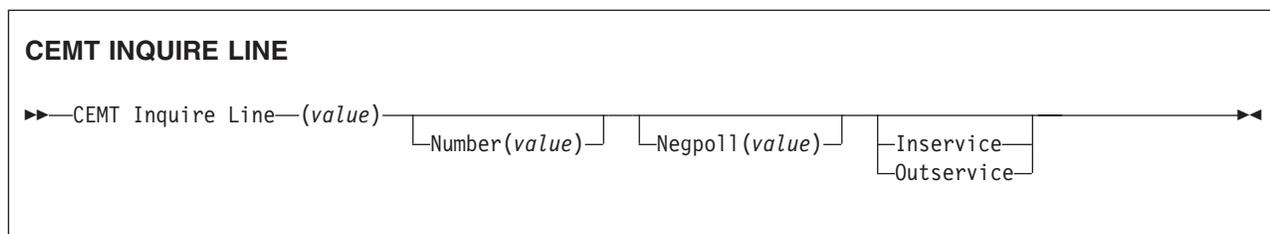
**Note:** There are blank fields on the screen where a value does not apply or is ‘negative’; that is, the value begins with ‘No’. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overwrite with input valid for that field. You may find it 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 122.

```
INQ LINE(TTT1)
RESULT
Line(TTT1)
Number(
Negpol( 00080 )
Servstatus( Inservice )
Purgetype(          )
```

Figure 122. 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

### Function

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 overwrite them with the required values.

*(value)*

specifies one or more names (1-8 characters) of a group of sessions.

ALI

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 123. CEMT INQUIRE MODENAME screen

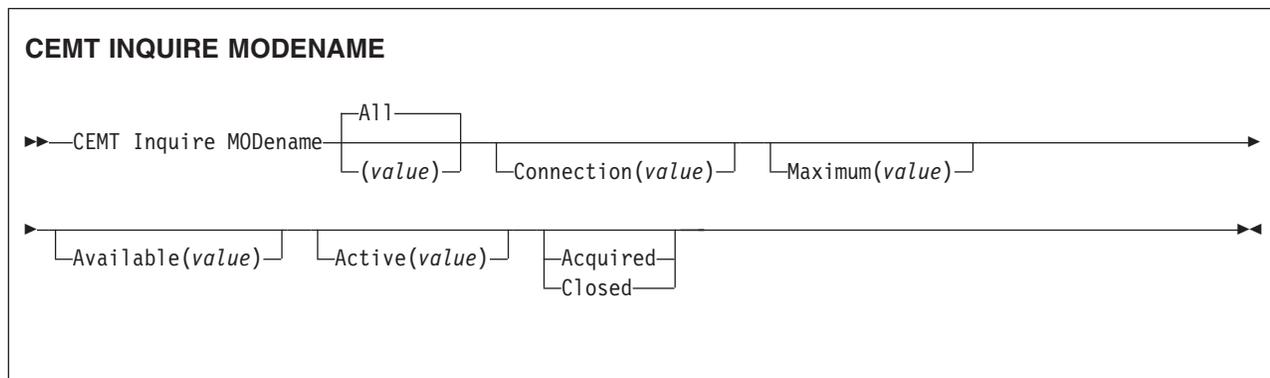
**Note:** There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overwrite with input valid for that field. You may find it 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 124 on page 264.

## CEMT INQUIRE MODENAME

```
I MODE
RESULT - OVERTYPE TO MODIFY
Modename(MODE1)
Connection( C )
Maximum(001)
Available( 000 )
Active(000)
Acqstatus( )
```

Figure 124. The expanded display



### 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

### Function

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.

### Sample screen

```
IN MONITOR
STATUS: RESULTS - OVERTYPE TO MODIFY
Mon On  Per Exc Res      Fre( 000000 ) Fil(16) Tsq(08)
App      Loc Sub(IYK2)
```

Figure 125. CEMT INQUIRE MONITOR screen

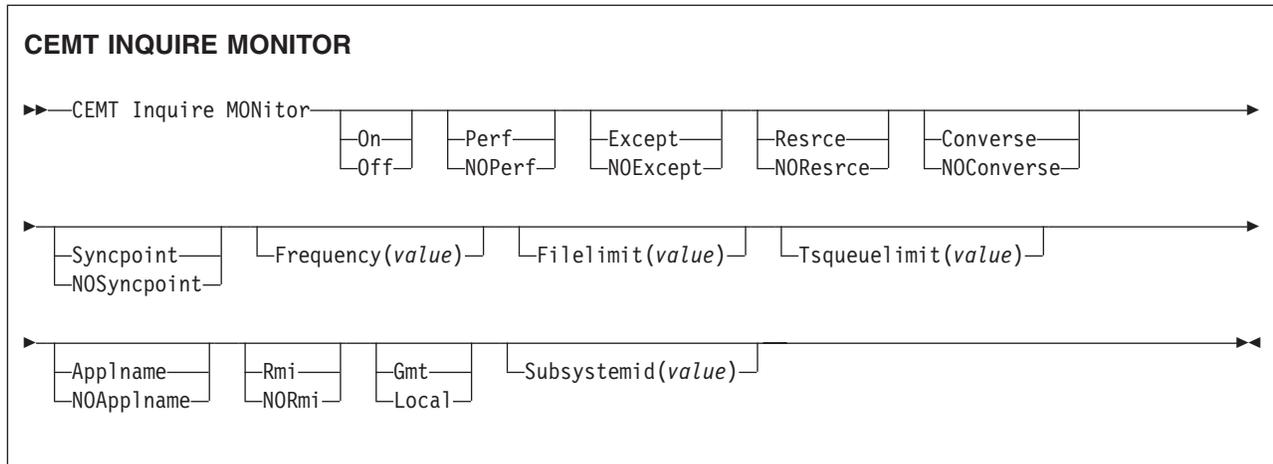
**Note:** There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it 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 126.

```
INQ MON
RESULT - OVERTYPE TO MODIFY
Monitor
Status( On )
Perfclass( Perf )
Exceptclass( Except )
Resrceclass( Resrce )
Conversest( Noconverse )
Syncpointst( Nosyncpoint )
Frequency( 000000 )
Filelimit(16)
Tsqueuelimit(08)
Applnamest(Applname)
Rmist(Normi)
Time(Local)
Subsystemid(IYK2)
```

Figure 126. The expanded display

## 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.

#### 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.

**Subsystemid (value)**

displays a 4-character name used as the subsystem identification in the MVS workload activity reports. The subsystem identification defaults to the first 4 characters of the VTAM generic APPLID.

For information on the MVS workload activity reports, see the *CICS Performance Guide*.

**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 EXEC CICS COLLECT STATISTICS MONITORING(taskno) command. The values are:

**Gmt** The time stamp fields in the performance class data are Greenwich mean time (GMT).

## CEMT INQUIRE MONITOR

**Local**            The time stamp fields in the performance class data are local time.

For programming information on the EXEC CICS COLLECT STATISTICS command, see the *CICS System Programming Reference*.

**TSqueue limit**

displays the maximum number of temporary storage queues for which transaction resource class monitoring data is being collected.

## CEMT INQUIRE NETNAME

### Function

Retrieve information about a network.

### Description

If you are familiar with network names, you may 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 will show you the details of only those netnames that are in service and available for use.

You can then tab to the highlighted or blank fields and overtype them with the required values.

(*value*)

specifies an 8-character network name.

AL1

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 127. 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 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).

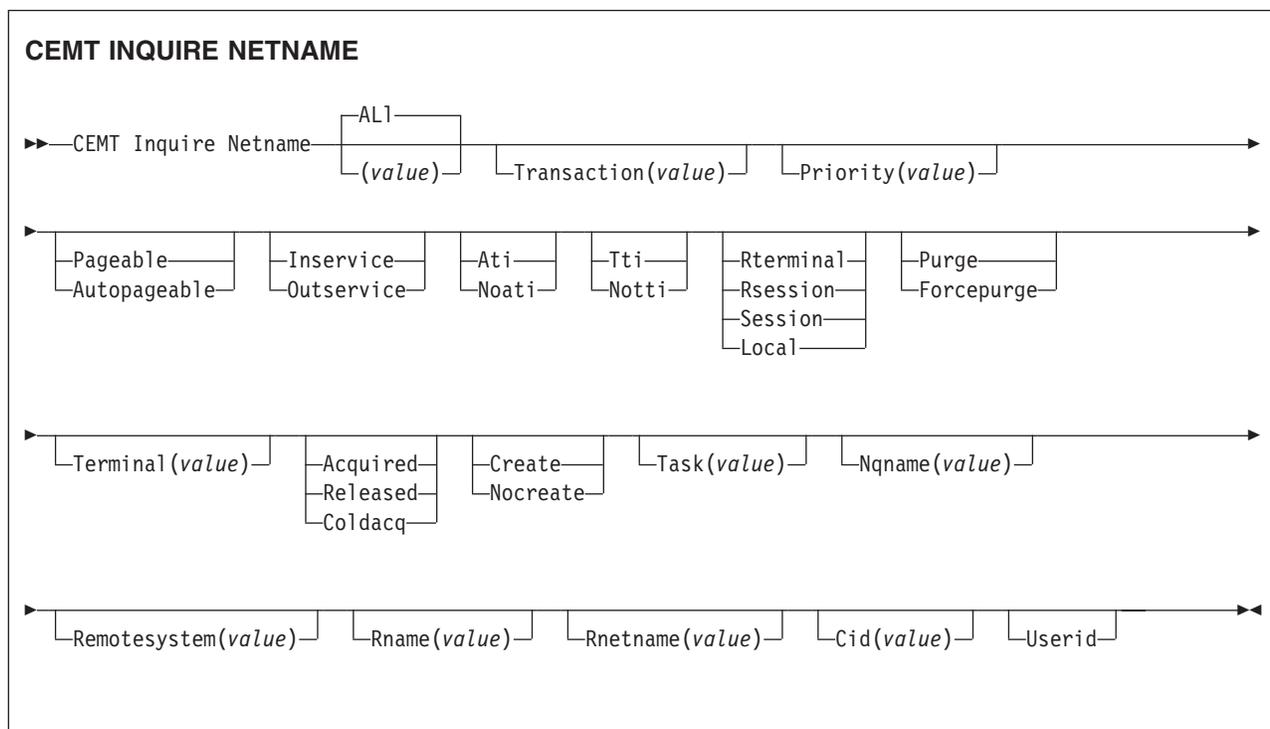
If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 128 on page 270.

## CEMT INQUIRE NETNAME

```

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( Ncreate )
Task(00033)
Remotesystem()
Nqname(GBIBMIYA.IYCQTC03)
Rname()
Rnetname()
Cid()
Userid()
    
```

Figure 128. The expanded display of an individual entry



### Displayed fields

#### 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.

**Notes:**

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.

## CEMT INQUIRE NETNAME

- 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:

## CEMT INQUIRE NETNAME

**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 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.

**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.

**Notes:**

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

### Function

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.

A11

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 129. 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 130.

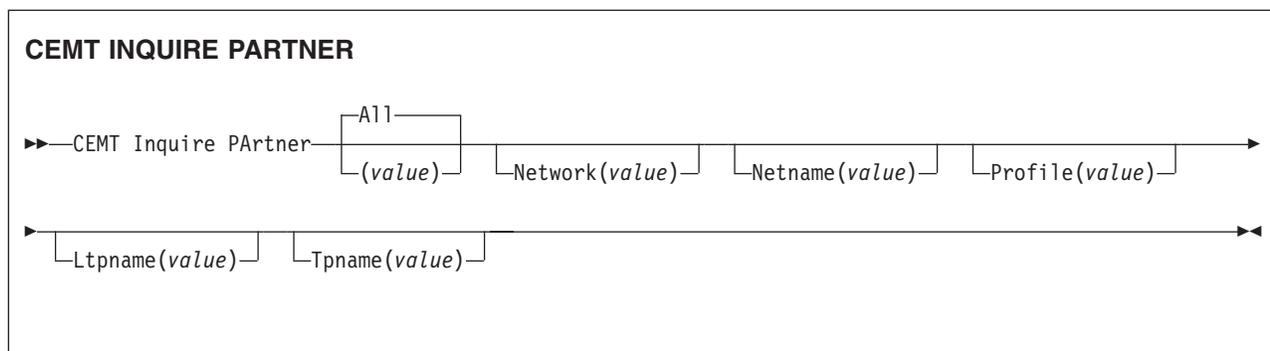
```

I PART
RESULT
  Partner(IST1ISC2)
  Network()
  Netname(IYMVM2)
  Profile(DFHCICSA)
  Ltpname(04)
  Tpname(IST1)

```

Figure 130. 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

### Function

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:

- Overtyping your changes on the INQUIRE screen after tabbing to the appropriate field.
- Use the CEMT SET PIPELINE command.

### ALI

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.

## CEMT INQUIRE PIPELINE

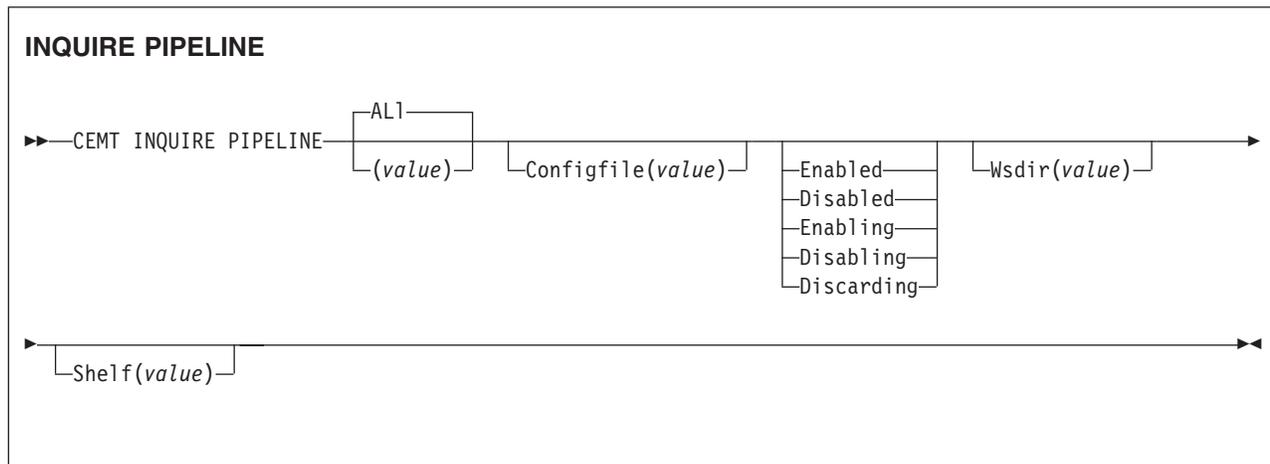
### Sample screen

```
I PIPELINE
STATUS: RESULTS - OVERTYPE TO MODIFY
Pip(LID852 ) Ena Con(/u/dbear1/pipeline/testru)
  She(/u/dbear1/shelf/ ) Wsd(/u/dbear1/ )
Pip(SAMPLE ) Ena Con(/u/dbear1/pipeline/testru)
  She(/u/dbear1/sampbind/ )

RESPONSE: NORMAL
PF 1 HELP      3 END      5 VAR      7 SBH 8 SFH 9 MSG 10 SB 11 SF

SYSID=CIAP APPLID=IYK2ZAF1
TIME: 09.13.28 DATE: 09.02.05
```

Figure 131. CEMT INQUIRE PIPELINE screen



**Conditions:** NOTAUTH, NOTFND

### Displayed fields

#### **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.

| **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.

| **Shelf**(*value*)

| Displays the name of the *shelf directory*. The name can be up to 255 characters  
| long.

## CEMT INQUIRE PROCESSTYPE

### Function

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:

- Overtyping your changes on the INQUIRE screen after tabbing to the appropriate field.
- Use the CEMT SET PROCESSTYPE command.

### AL1

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 132. CEMT INQUIRE PROCESSTYPE screen

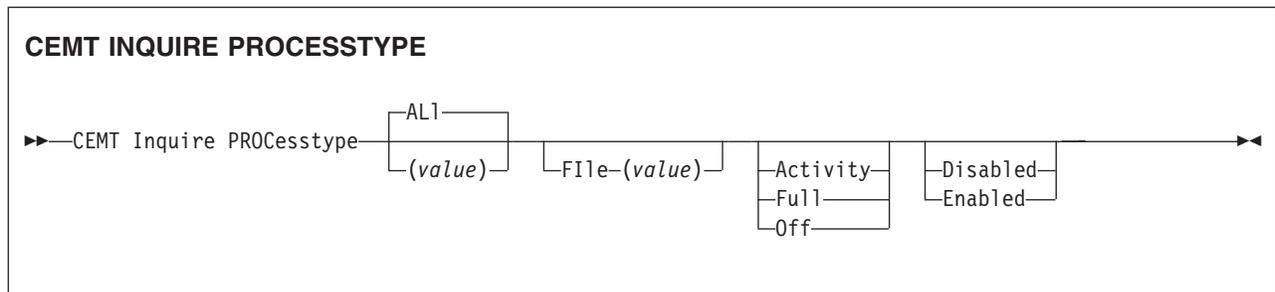
If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 133 on page 281.

```

I PROC
STATUS:  RESULTS - OVERTYPE TO MODIFY
  Processtype(SALESTYPE4 )
  File(SALESF4 )
  Enablestatus( Enabled )
  Auditlog(PLOG51 )
  Auditlevel( Activity )

```

Figure 133. The expanded display of an individual entry



## 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* manual.

### 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

## CEMT INQUIRE PROFILE

### Function

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.

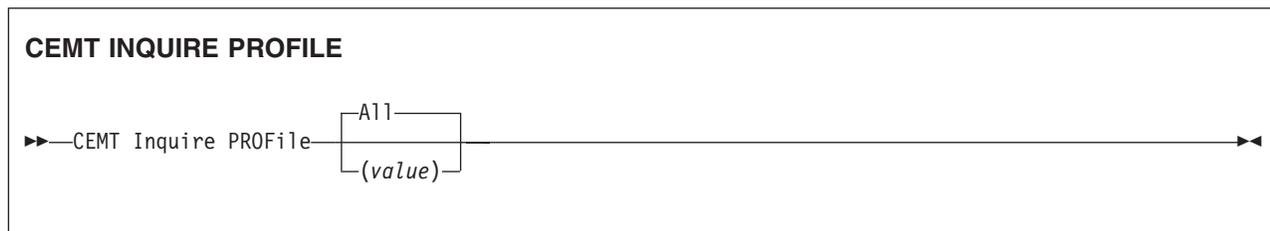
A11

is the default.

### Sample screen

```
IN PROFILE
STATUS: RESULTS
Prof(DFHCICSA)
Prof(DFHCICSE)
Prof(DFHCICSF)
Prof(DFHCICSR)
Prof(DFHCICSS)
Prof(DFHCICST)
Prof(DFHCICSV)
```

Figure 134. CEMT INQUIRE PROFILE screen



### 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

### Function

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:** JVM programs 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 overwrite 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.

**AL1**

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 "clasicid".

If the clasicid specified is the suffix of a PLT that does not have a program definition, the INQUIRE PROGRAM CLASS(clasicid) 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(00000000) Bel Uex Ful
Prog(DFH$AALL) Len(0000002936) Ass Pro Ena Pri Ced
Res(000) Use(00000000) Bel Uex Ful
Prog(DFH$ABRW) Len(0000002520) Ass Pro Ena Pri Ced
Res(000) Use(0000000008) Bel Uex Ful Qua Nat
Prog(DFH$ABRW) Len(0000002520) Ass Pro Ena Pri Ced
Res(000) Use(00000001) Bel Uex Ful
Prog(DFH$ACOM) Len(0000001128) Ass Pro Ena Pri Ced
Res(000) Use(00000000) Bel Uex Ful
Prog(DFH$AGA ) Len(0000000520) Map Ena Pri
Res(000) Use(00000002)
Prog(DFH$AGB ) Len(0000000448) Map Ena Pri
Res(000) Use(00000000)
Prog(DFH$AGC ) Len(0000000512) Map Ena Pri
Res(000) Use(00000000)
Prog(DFH$AGD ) Len(0000000576) Map Ena Pri
Res(000) Use(00000000)
+ Prog(DFH$AGK ) Len(0000000392) Map Ena Pri
Res(000) Use(00000000)

```

Figure 135. 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 136 on page 285 and Figure 137 on page 285.

```

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(00000000)
  Dataloc(Below)
  Execkey(Uexeckey)
  Executionset( Fullapi )
  Concurrency(Threadsafe)
  Apist(Cicsapi)
  Remotesystem()
  Runtime( Jvm )
    
```

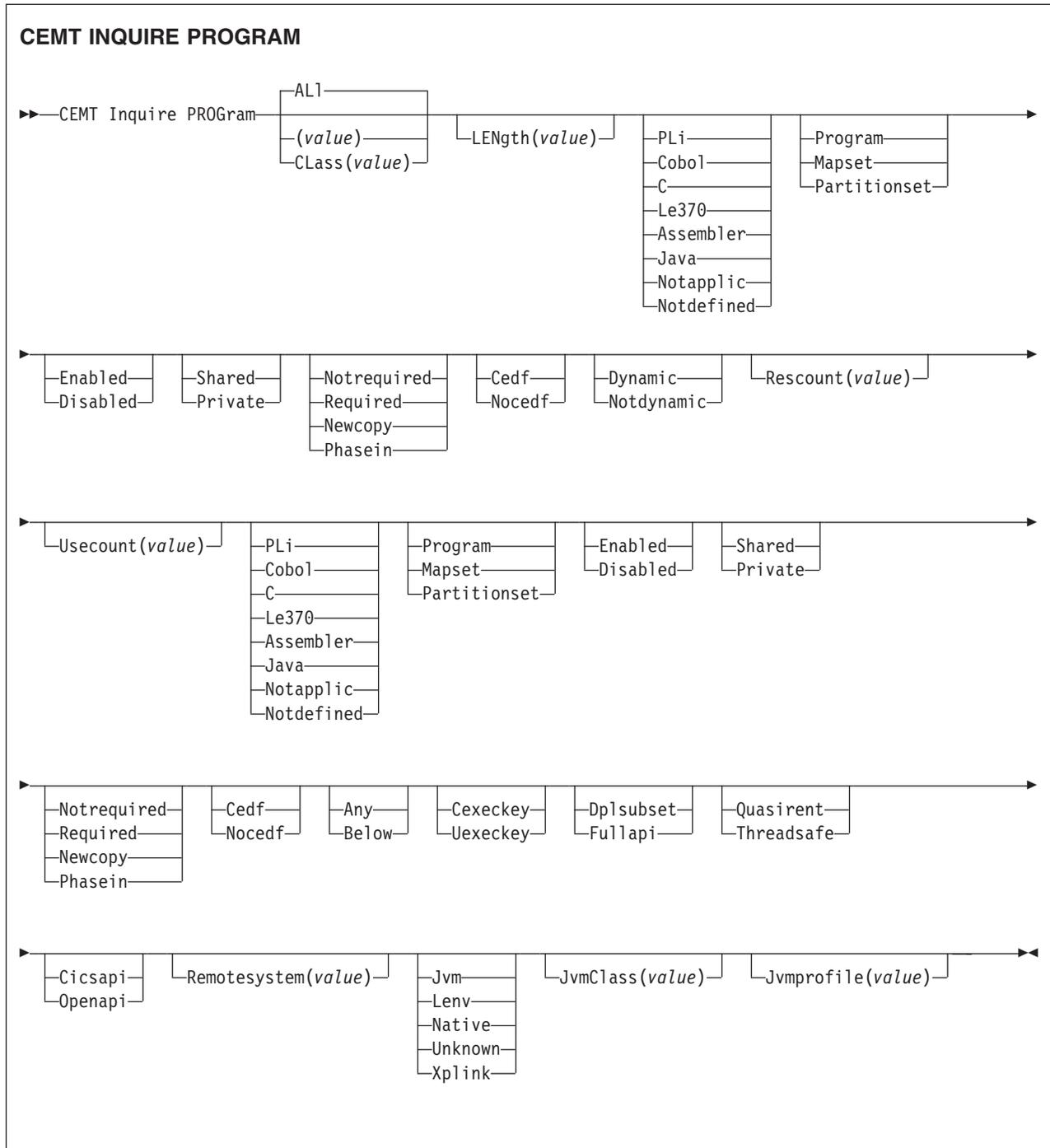
Figure 136. The expanded display of an individual entry (first panel)

```

IN PROG
RESULT - OVERTYPE TO MODIFY
  Program(DFH$AALL)
  Runtime( Jvm )
  Jvmclass( JavaApp1ClassFile      )
  Jvmclass(                          )
  Jvmclass(                          )
  Jvmclass(                          )
  Jvmclass(                          )
  Jvmprofile(DFHJVMPR)
    
```

Figure 137. 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 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 DFHRPL library concatenation.

You can reset this value by overtyping it. The values are:

**Newcopy** CICS uses a new copy of the program when all the transactions currently using the program have finished (RESCOUNT equal to zero). CICS loads the new version either from the DFHRPL library concatenation or uses an LPA-resident version, depending on the PRIVATE or SHARED options. PRIVATE is the default setting.

## CEMT INQUIRE PROGRAM

**Note:** NEWCOPY cannot be specified for a program specified with the HOLD option or for a JVM program.

**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 DFHRPL 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.

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 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 class specified on the program definition, up to a maximum of 255 characters.

**Jvmprofile**

displays the name of the JVM profile associated with this program.

**Language**

displays the language in which the program is written. The values are:

**Pli, Cobol, C, Le370, or Assembler**

Supported languages. In the case of LE370, the program exploits multi-language support, or has been compiled by the SAA Language Environment/370 (LE/370)-enabled 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 JVM program.

**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.

## CEMT INQUIRE PROGRAM

**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 JVM program.

**Note:** This is not necessarily the same as the number of times the module has been loaded from the CICS load library (DFHRPL).

### **Runtime**

displays information about the runtime environment of the program. The values are:

**Jvm** The program is a JAVA program running under the control of a 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 is 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.

**Private** The next new copy of the program is to be loaded from the DFHRPL library.

**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. No value is returned if the module is a remote program, or a JVM program.

## CEMT INQUIRE REQUESTMODEL

### Function

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.

AL1

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 138. CEMT INQUIRE REQUESTMODEL screen

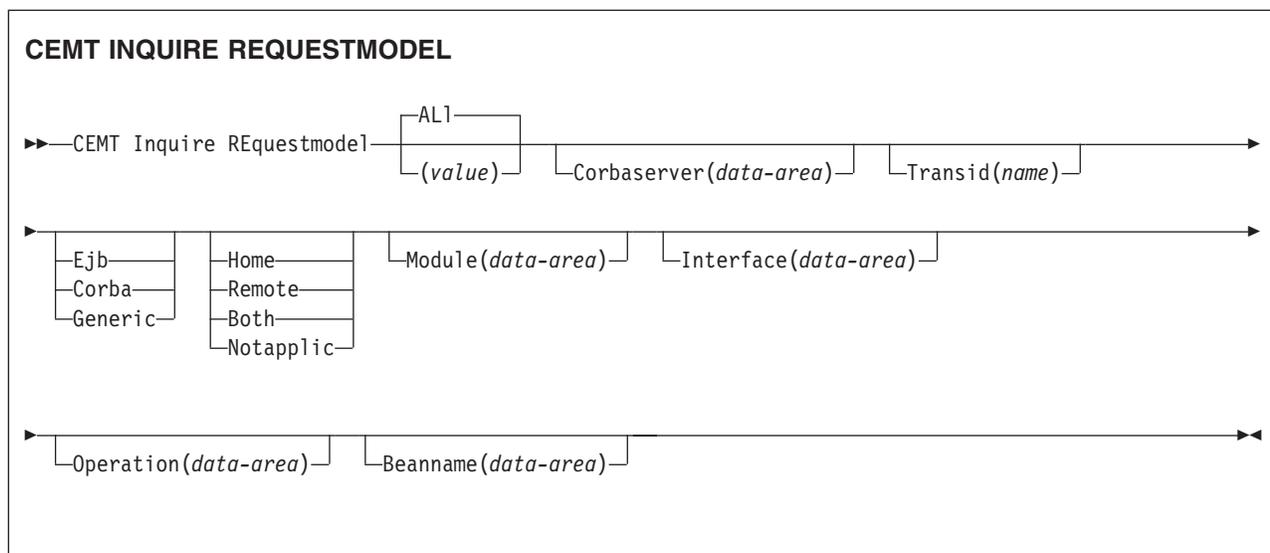
If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format.

```

I RE
RESULT - OVERTYPE TO MODIFY
All
Requestmodel(DFHGFACT)
Corbaserver
Transid
Type
Intfacetype
Module
Interface
Operation
Beaname
  
```

Figure 139. 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.

#### **Beaname**(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

## CEMT INQUIRE RRMS

### Function

Retrieve RRMS information.

### Description

INQUIRE RRMS tells you if CICS does or does not accept inbound transactional EXCI work.

### Input

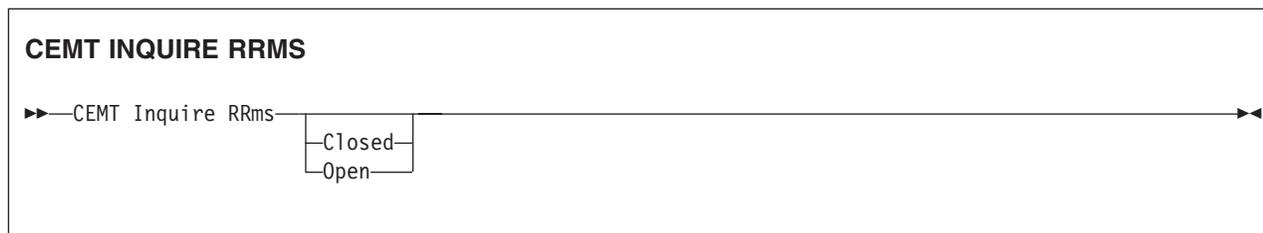
Press the Clear key and type CEMT INQUIRE RRMS (the minimum abbreviation is CEMT I R).

You will get a display screen, a sample of which is shown below.

### Sample screen

```
IN RRMS
STATUS: RESULTS
Rrm Ope
```

Figure 140. CEMT INQUIRE RRMS screen



### 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

### Function

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 141. 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 142 on page 296.

## CEMT INQUIRE STATISTICS

```
IN STAT
RESULT - OVERTYPE TO MODIFY
Statistics
Recording( Off )
Record(          )
Reset(          )
Interval( 030000 )
Endofday( 000000 )
Nexttime(000000)
```

Figure 142. The expanded display

### CEMT INQUIRE STATISTICS

▶▶—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.

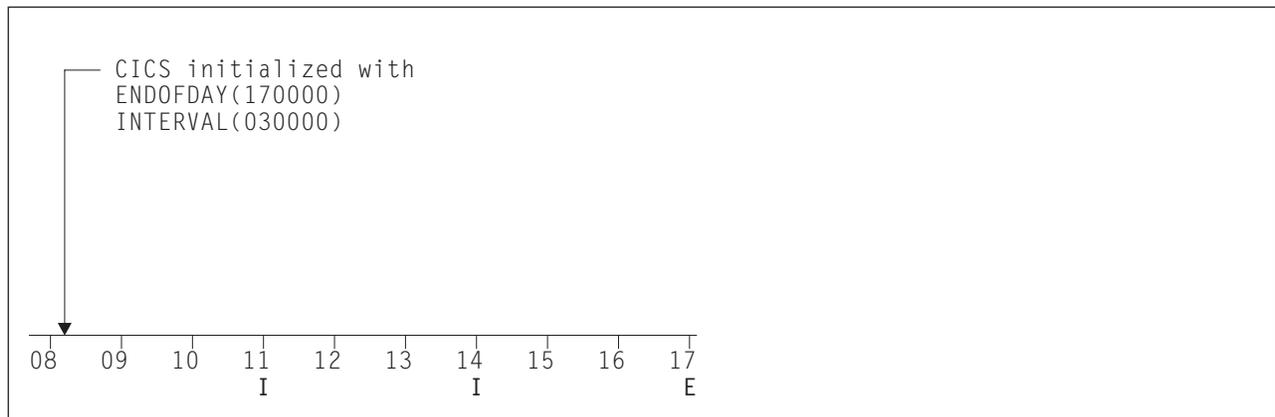


Figure 143. Changing the INTERVAL or ENDOFDAY value

**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.

**Notes:**

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.

## CEMT INQUIRE STATISTICS

**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

### Function

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 overwrite 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.

AL1

is the default.

### Sample screen

```
IN STRE
STATUS: RESULTS
Str(BELL.CICSHT61.DFHLOG      ) Use(0000001) Ok Sys
```

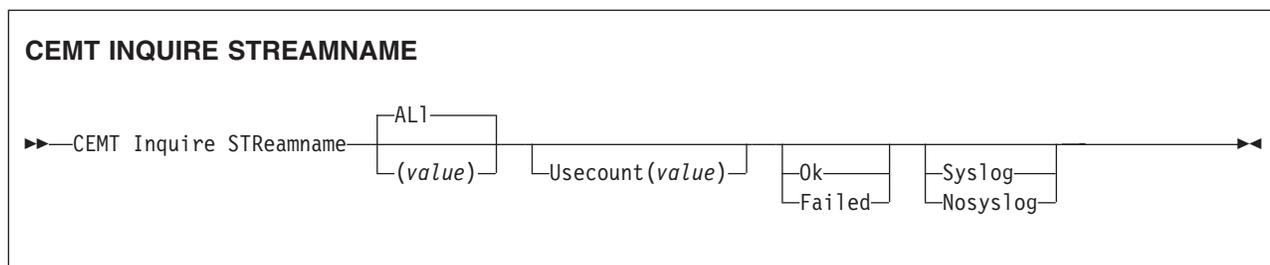
Figure 144. CEMT INQUIRE STREAMNAME screen

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 145.

```
IN STRE
RESULT
Streamname(BELL.CICSHT61.DFHLOG)
Usecount(0000001)
Status(Ok)
Systemlog(Syslog)
```

Figure 145. 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

### Function

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 the *CICS Messages and Codes* manual 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,

## CEMT INQUIRE SYDUMPCODE

for example, if you enter `cent 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 overwrite them with the required values.

*(value)*

is an 8-character system dump code. See the *CICS Messages and Codes* manual for details of the codes.

**AL1**

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)
```

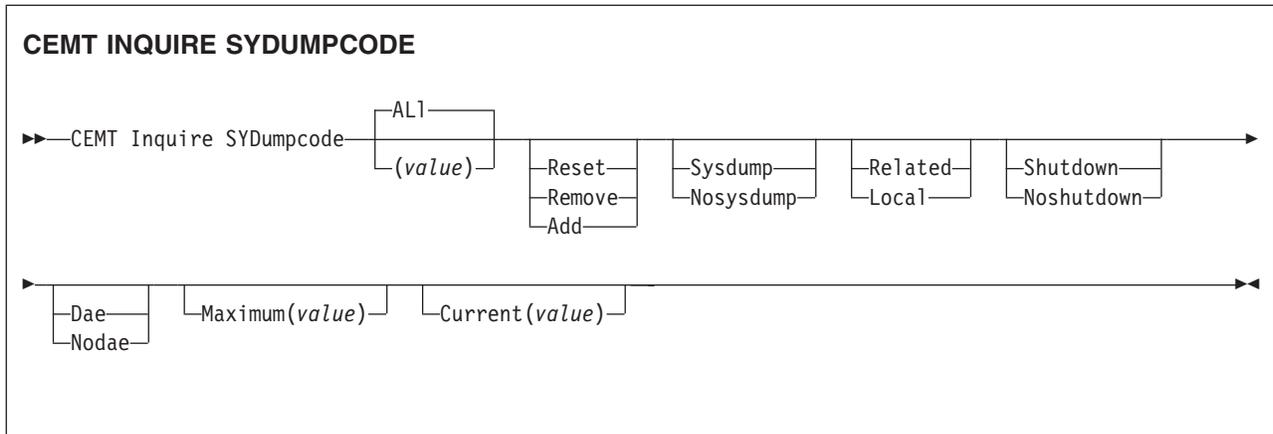
Figure 146. CEMT INQUIRE SYDUMPCODE screen

**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 overwrite 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 147.

```
I SYD
RESULT - OVERTYPE TO MODIFY
Sydumpcode(AP0001)
Sysdumping( Sysdump )
Dumpscope( Local )
Shutoption( Noshutdown )
Maximum( 010 )
Current(0000)
Action(          )
Daeoption( Dae )
```

Figure 147. 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.

## CEMT INQUIRE SYDUMPCODE

**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.

### Sydumrcode (*value*)

indicates that this panel relates to a SYDUMPCODE inquiry and displays an 8-character system dump code. See the *CICS Messages and Codes* manual 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

### Function

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 overtyping them with the required values.

To change various attributes, you can:

- Overtyping your changes on the INQUIRE screen after tabbing to the appropriate field (see Figure 148)
- Use the CEMT SET SYSTEM command.

### Sample screen

```

inq system
STATUS: RESULTS - OVERTYPE TO MODIFY
Aging( 32768 )           Progautoexit( DFHPGADX )
Akp( 04000 )           Progautoinst( Autoactive )
Cicstslevel(030100)     Reentprotect(Reentprot)
Cmdprotect(Nocmdprot)  Release(0640)
Db2conn()              Runaway( 0005000 )
Debugtool( Nodebug )   Scandelay( 0500 )
Dfltuser(WALSH)        Sdtran(CESD)
Dsalimit( 07340032 )   Sosstatus(Notsos)
Dsrtprogram( NONE )    Storeprotect(Active)
Dtrprogram( DFHDYP )   Time( 0001000 )
Dumping( Sysdump )     Tranisolate(Active)
Edsalimit( 0268435456 )
Forceqr( Noforce )
Logdefer( 00030 )
Maxtasks( 075 )
Mrobatch( 001 )
Oslevel(010400)
Progautoctlg( Ctlgmodify )

                                     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 148. CEMT INQUIRE SYSTEM screen

### CEMT INQUIRE SYSTEM

▶▶—CEMT Inquire SYstem—▶▶

### Displayed fields

#### Aging (value)

displays the time factor for CICS to use in the priority aging algorithm for

## CEMT INQUIRE SYSTEM

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.

### **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 OS/390® product under which the CICS region is running. The value is of the form *vrrmm*, and CICS TS Version 3 Release 1 returns 030100.

### **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:

- |                  |  |
|------------------|--|
| <b>Cmdprot</b>   | 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.             |
| <b>Nocmdprot</b> | 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. (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.

**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. (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.

**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.

## CEMT INQUIRE SYSTEM

### **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 SIT 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.

#### **Notes:**

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.

### **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:

- |                   |   |
|-------------------|---|
| <b>Ctlgall</b>    | All autoinstalled program definitions are to be cataloged and restored on a warm or emergency start.  |
| <b>Ctlgmodify</b> | 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. |
| <b>Ctlgnone</b>   | 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:

- |                     |  |
|---------------------|--|
| <b>Autoactive</b>   | Autoinstall for programs is active. On first use, if a program, mapset, or partitionset is not defined, the definition is created dynamically. |
| <b>Autoinactive</b> | 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 0640 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 OS/390, 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.

**Notes:**

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.

## CEMT INQUIRE SYSTEM

### **Sosstatus** (*value*)

displays whether CICS is short on storage in any of the dynamic storage areas. The values returned are:

<b>Notsos</b>	CICS is not short on storage in any of the dynamic storage areas.
<b>Sosabove</b>	CICS is short of storage in at least one of the dynamic storage areas above 16MB.
<b>Sosbelow</b>	CICS is not short on storage in any of the dynamic storage areas below 16MB.
<b>Sos</b>	CICS is short of storage in at least one of the dynamic storage areas.

### **Storeprotect** (*value*)

displays whether storage protection is active in the CICS region. The values returned are:

<b>Active</b>	CICS is operating with the CICS storage protection facility.
<b>Inactive</b>	CICS is operating without the CICS storage protection facility.

See the description of the STGPROT system initialization parameter in 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'.

#### **Notes:**

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:

<b>Active</b>	Transaction isolation is active in the CICS region.
<b>Inactive</b>	CICS is running without transaction isolation, either because the support is not available, or it was not requested at CICS initialization.

## CEMT INQUIRE TASK

### Function

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:

- Overtyping your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 140).
- Use the CEMT SET TASK command.

*(value)*

is the CICS-generated task number, in the range 1–99999.

**AL1**

is the default. The maximum number of tasks displayed is 32000.

**TC1** *(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(B0C9D8E9C3B1FB09)
```

Figure 149. 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,

## CEMT INQUIRE TASK

(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 150.

```
I TASK
RESULT - OVERTYPE TO MODIFY
Task(0000026)
Tranid(CEMT)
Facility(S262)
Runstatus(Suspended)
Ftype(Term)
Priority( 255 )
Purgetype(          )
Startcode(T0)
Userid(CICSUSER)
Uow(B0C9D8D111440007)
Htype(ZC10WAIT)
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 150. The expanded display of an individual entry



## CEMT INQUIRE TASK

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:

<b>Task</b>	The task was initiated from another task.
<b>Term</b>	The task was initiated from a terminal.
<b>Dest</b>	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:

<b>Backout</b>	All changes made to recoverable resources are to be backed out.
<b>Commit</b>	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:

- |               |  |
|---------------|--|
| <b>Nowait</b> | 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. |
| <b>Wait</b>   | 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 MQ™ correlator for the 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:

- |                   |   |
|-------------------|---|
| <b>Purge</b>      | The task is to be terminated. Termination occurs only when system and data integrity can be maintained.   |
| <b>Forcepurge</b> | 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:

- |                     |                           |
|---------------------|---------------------------|
| <b>Running</b>      | The task is running.      |
| <b>Dispatchable</b> | The task is dispatchable. |
| <b>Suspended</b>    | The task is suspended.    |

**Startcode** (*value*)

displays how this task was started. The values are:

- |           |   |
|-----------|---|
| <b>D</b>  | A distributed program link (DPL) request. The program cannot issue I/O requests against its principal facility or any syncpoint requests. |
| <b>Ds</b> | A distributed program link (DPL) request, as for code D, with the exception that the program can issue syncpoint requests.                |

## CEMT INQUIRE TASK

<b>Qd</b>	A transient data trigger level was reached.
<b>S</b>	Start command (no data)
<b>Sd</b>	Start command (with data)
<b>To</b>	The operator typed a transaction code at the terminal.
<b>Tp</b>	The transaction was started by presetting the transaction ID for the terminal.
<b>U</b>	User-attached task.

# **Note:** When the IIOOP request processor is run locally the startcode for an  
# ASSIGN command or an INQUIRE TASK is U. When the IIOOP request  
# processor is run remotely, over an MRO link, the startcode for these  
# commands is TO. (If you attempt to run the IIOOP 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 1–999999.

### **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 master JVM (JM) TCB
- The resource-owning mode (RO) TCB
- The ONC/RPC mode (RP) TCB
- The sockets listener mode (SL) TCB
- The secure sockets layer mode (SO) TCB
- A sockets mode (S8) TCB
- The FEPI mode (SZ) TCB.

**Qr** The task is running under the CICS QR TCB.

**Ukopen** The task is running under a user key open TCB (for example, a J9 mode TCB, 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

### Function

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.

AL1

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)

```

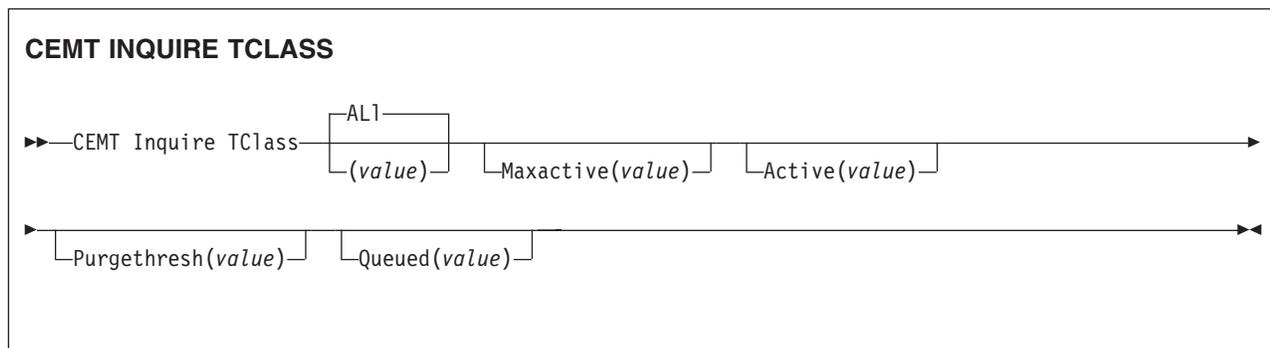
Figure 151. CEMT INQUIRE TASK screen

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 152 on page 318.

## CEMT INQUIRE TCLASS

```
IN TCLASS
RESULT - OVERTYPE TO MODIFY
Tclass(DFHTCIND)
Maxactive( 010 )
```

Figure 152. The expanded display of an individual entry



### 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

### Function

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:

- Overtyping your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 140).
- Use the CEMT SET TCPIP command.

### Displayed fields

#### Actsockets(*value*)

displays the current number of active IP sockets managed by the CICS sockets domain.

#### Cr1profile(*value*)

Displays the name of the profile that is used to authorize CICS to access the certificate revocation lists that are stored 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.

#### Ss1cache(*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 TCPIP SERVICE

### Function

Retrieve information about TCP/IP ports on which CICS internal TCP/IP support is currently listening on behalf of other CICS services.

### Description

INQUIRE TCPIP SERVICE 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 TCPIP SERVICE (the minimum abbreviation is CEMT I TCPIPS). You get a display that lists the current status.
- Type CEMT INQUIRE TCPIP SERVICE (CEMT I TCPIPS) followed by as many of the other attribute settings that you wish to view.

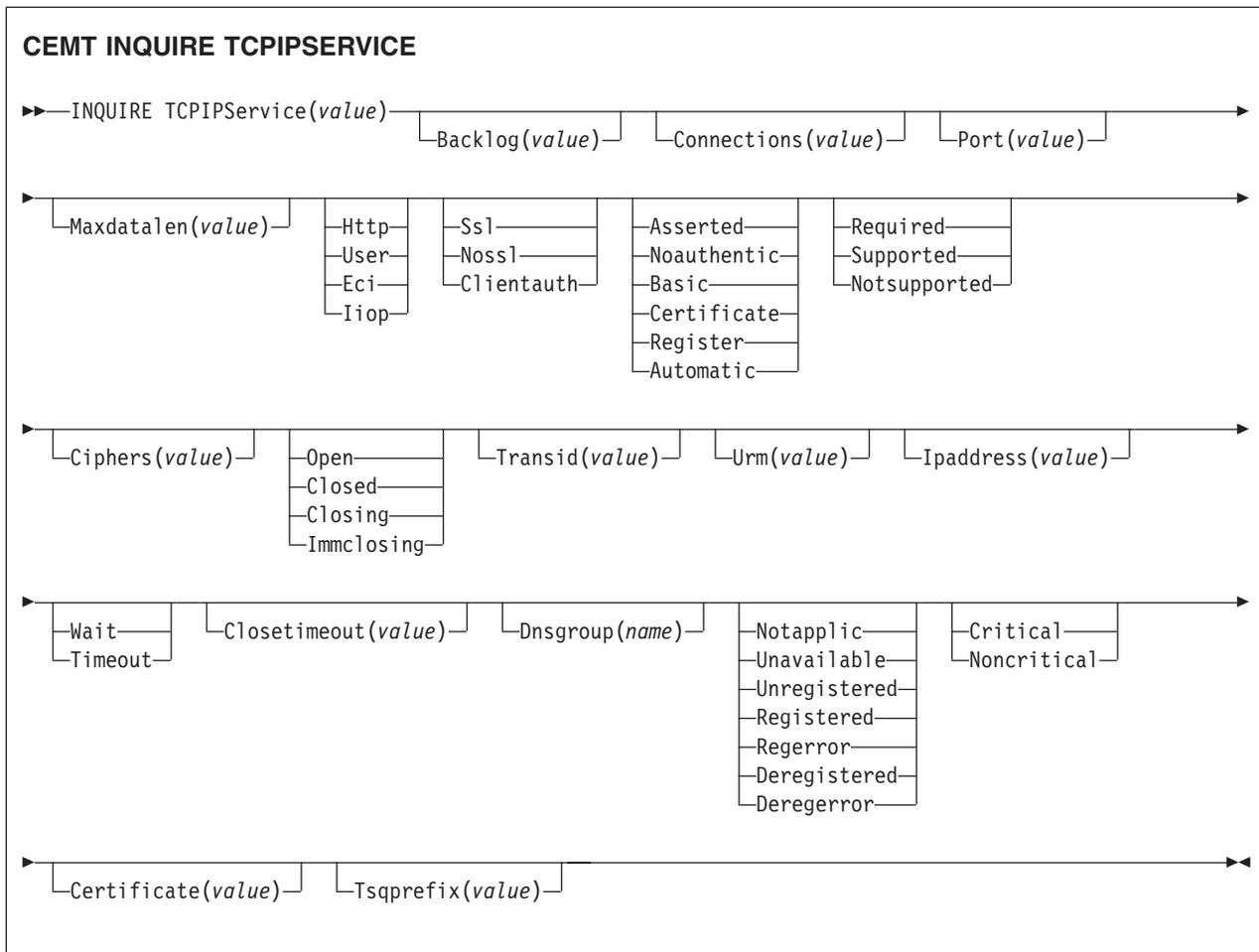
To change various attributes, you can:

- Overtyping your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 140).
- Use the CEMT SET TCPIP SERVICE command.

### Sample screen

```
inq tcpip service
RESULT - OVERTYPE TO MODIFY
  Tcpservice(HTTPSSL)
  Backlog( 00005 )
  Connections(00000)
  Port(20130)
  Maxdatalen( 000032 )
  Protocol(Http)
  Ssltype(Ssl)
  Authenticate(Noauthentic)
  Privacy(Supported)
  Ciphers(0504352F0A0903060201)
  Openstatus( Open )
  Transid(CWXN)
  Urm( DFHWBADX )
  Ippaddress(9.20.101.6)
  Socketclose(Timeout)
  Closetimeout(000005)
  Dnsgroup()
  Dnsstatus(          )
  Grpcritical(Noncritical)
  Certificate()
  Tsqprefix()
```

Figure 153. CEMT INQUIRE TCPIP SERVICE screen



## Displayed fields

### 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.

## CEMT INQUIRE TCPIPSERVICE

This value is returned only when PROTOCOL has a value of HTTP or IIOF.

### **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 IIOF 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 OS/390.

**Unregistered**

Registration has not yet occurred (this is the initial state of any service).

**Registered**

Registration has completed successfully.

**Regerror**

Registration has failed with an error.

**Deregistered**

Deregistration has completed successfully.

**Deregerror**

Deregistration has failed with an error.

**Grpcritical**

returns a CVDA value specifying whether or not this TCPIP SERVICE is a critical member of the DNS group. The CVDA values are:

**Critical**

If this TCPIP SERVICE is closed, or abnormally stops listening for any reason, the group name specified in the DNSGROUP attribute is deregistered from WLM.

**Noncritical**

If this TCPIP SERVICE 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.

**Ippaddress (value)**

returns an IP address. If you specify an IP address in the TCPIP SERVICE definition, that address is returned, otherwise the default IP address is returned. If there is more than one IP address on an OS/390 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 3K through 524288K. The default is 32K.

**Openstatus**

returns the state of the service. 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.

**Port (value)**

returns the number of the port on which CICS is listening on behalf of this service.

## CEMT INQUIRE TCPIP 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.

#### Not supported

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.

#### User

The user-defined protocol is used. Requests are passed to the analyzer program for the TCPIP SERVICE, and handled using CICS Web support facilities, but the HTTP specifications are not used to check the messages.

#### Iiop

Connections are handled by CICS IIOPI support. IIOPI is required for TCPIP SERVICES that are to accept inbound requests for enterprise beans.

### Socketclose

returns a CVDA value indicating whether a timeout value is in effect for the TCPIP SERVICE. CVDA values are:

#### Wait

**NO** was specified on the definition. Socket receives will wait for data indefinitely.

#### Timeout

A value was specified for the SOCKETCLOSE parameter on the definition.

### Ssltype

returns a CVDA value indicating the status of SSL support for this service. CVDA values are

#### Ss1

Non client authentication SSL flows are processed on this port

#### Noss1

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*)

returns the 6-character prefix used when the service generates TS queue names.

**Urm** (*value*)

8 character name of Service URM invoked by attached task.

## CEMT INQUIRE TDQUEUE

### Function

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:

- Overtyping your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 140).
- Use the CEMT SET TDQUEUE command.

(*value*)

is the identifier (1-4 characters) of a transient data queue.

A11

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 154. 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,

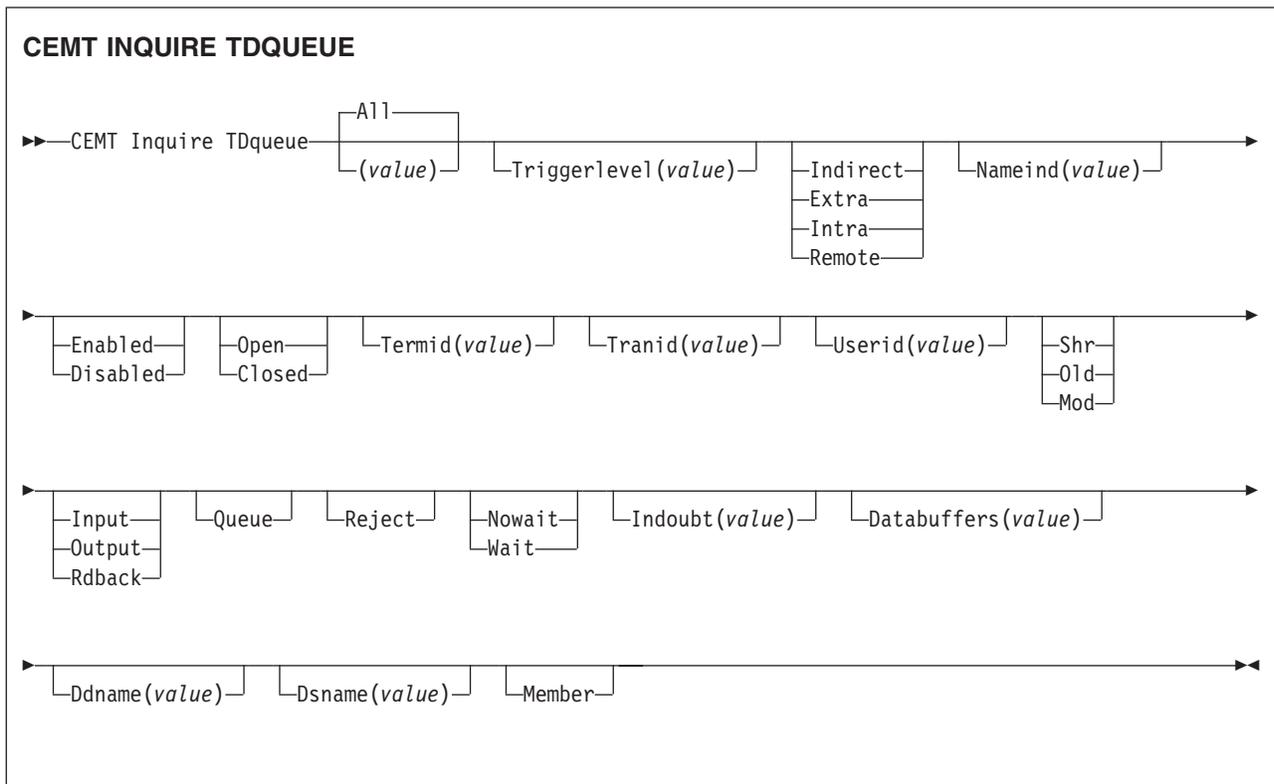
(negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 155.

```

IN TDQ
RESULT - OVERTYPE TO MODIFY
Tdqueue(CCS0)
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 155. The expanded display of an individual entry



**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.

## CEMT INQUIRE TDQUEUE

**Note:** An extrapartition queue can be input or output, but not both.

**Ddname** (*value*) (*extrapartition 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.

### Notes:

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. See the *CICS Messages and Codes* manual for an explanation of the interactions of in-doubt attributes on the TDQUEUE and TRANSACTION definitions.

**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.

**Notes:**

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 DELETEDQ requests
- WRITEQ, if the indoubt UOW has issued WRITEQ or DELETEDQ requests
- DELETEDQ, if the indoubt UOW has issued READQ, WRITEQ or DELETEDQ 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.

## CEMT INQUIRE TDQUEUE

### **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:

<b>Indirect</b>	The queue type is indirect. The name of the final target queue is shown in the next field.
<b>Extra</b>	The queue type is extrapartition.
<b>Intra</b>	The queue type is intrapartition.
<b>Remote</b>	The queue type is remote.

### **Userid**

displays the 8-byte identifier associated with the intrapartition queue intended for ATI.

## CEMT INQUIRE TERMINAL

### Function

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 may prefer to use the command that uses these names. See “CEMT INQUIRE NETNAME” on page 269.

The terminal entry displayed can also be an MVS console.

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.

### 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 current 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 `cent i te i at`, the resulting display will show you the details of only those terminals that are in service and available for use. Or if you enter `cent i te rem(cicr)`, the display will show you only 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:

- Overtyping your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 140).
- 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

CEMT INQUIRE TERMINAL(S201)

### AL1

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 1- or 2-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($225) Tra(CEMT) Pri( 000 ) Pag Ins Ati Tti Loc
Net(IGCS225 ) Acq Tas(0000068) Nqn(GBIBMIYA.IGCS225 )
Ter($280) 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(TEMPLATE1) Rel Rem(CBPS)
Ter(-AA0) Pri( 000 ) Aut Ins Ati Tti Ses
Net(TEMPLATE1) Rel Cre Rem(CBPS)
+ Ter(-AA1) Pri( 000 ) Aut Out Ati Tti Ses
Net(TEMPLATE1) Rel Cre Rem(CBPS)
```

Figure 156. CEMT INQUIRE TERMINAL screen

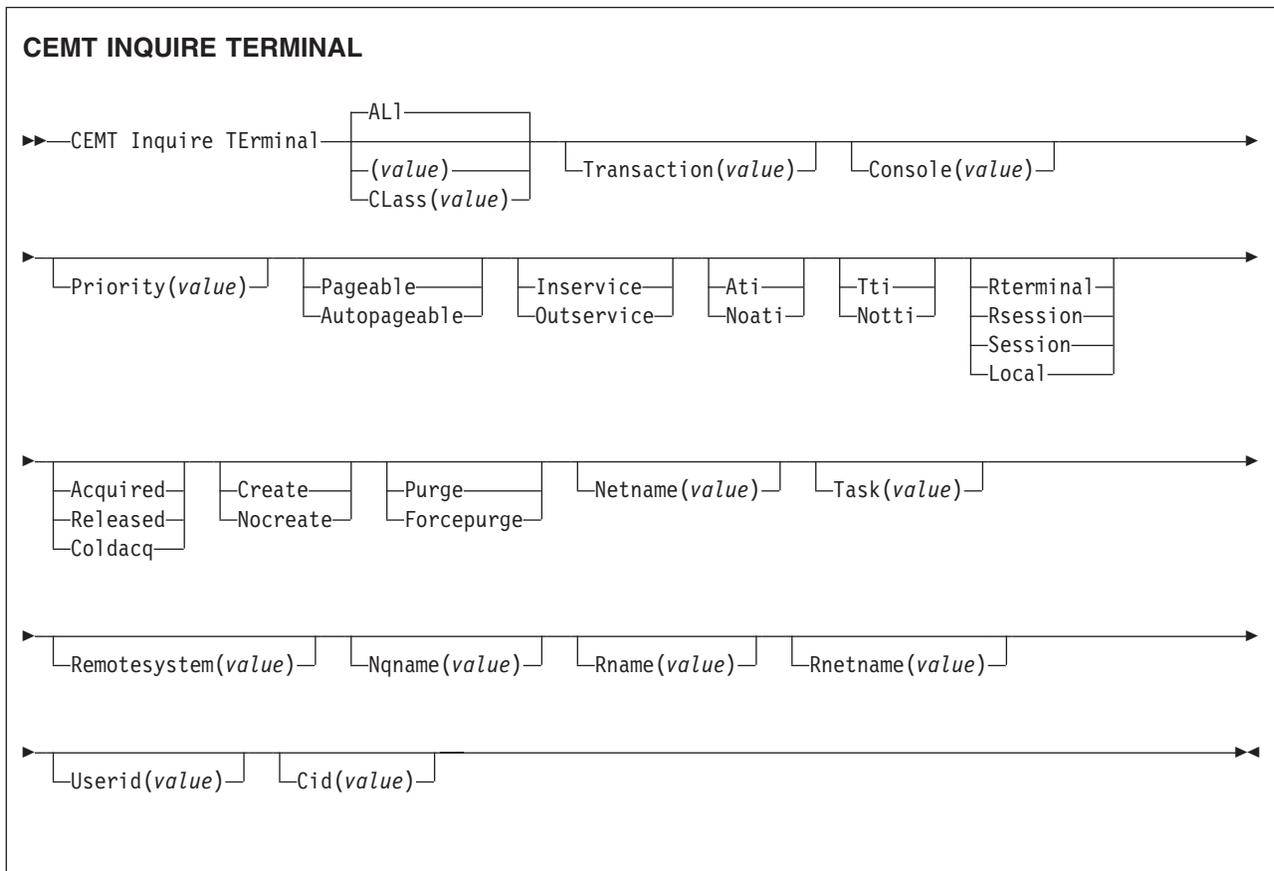
**Note:** There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it 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 157 on page 333.

```

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 157. The expanded display of an individual entry



### Displayed fields

#### Atistatus

displays whether the terminal is available for use by transactions that are

## CEMT INQUIRE TERMINAL

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.

### Notes:

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 it isn't.
- A period separating the name from the numeric ID.
- The last 3 bytes contain the numeric console ID.

### 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.

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:

<b>Rterminal</b>	The terminal is a remote device.
<b>Rsession</b>	The terminal is a session of a remote connection.
<b>Session</b>	The terminal is an ISC or IRC session.
<b>Local</b>	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 Transaction Server for z/OS 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:

<b>Pageable</b>	Pages are written on request.
<b>Autopageable</b>	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:

## CEMT INQUIRE TERMINAL

- Purge** Transactions will be 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 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 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 **Netname**.

**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.

**Notes:**

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

### CEMT INQUIRE TRANSACTION

#### Function

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:

- Overtyping your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 140).
- 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.

**AL1**

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$AALL) Tc1( DFHTCL00 ) Ena Pur
Prf(DFHCICST) Uda Bel Iso
Tra(ABRW) Pri( 001 ) Pro(DFH$ABRW) Tc1( DFHTCL00 ) Ena Pur
Prf(DFHCICST) Uda Bel Iso
Tra(ADYN) Pri( 001 ) Pro(DFH99 ) Tc1( DFHTCL00 ) Ena Pur
Prf(DFHCICST) Uda Bel Iso
Tra(AINQ) Pri( 001 ) Pro(DFH$AALL) Tc1( DFHTCL00 ) Ena Pur
Prf(DFHCICST) Uda Bel Iso
Tra(AMNU) Pri( 001 ) Pro(DFH$AMNU) Tc1( DFHTCL00 ) Ena Pur
Prf(DFHCICST) Uda Bel Iso
Tra(AORD) Pri( 001 ) Pro(DFH$AREN) Tc1( DFHTCL00 ) Ena Pur
Prf(DFHCICST) Uda Bel Iso
Tra(AORQ) Pri( 001 ) Pro(DFH$ACOM) Tc1( DFHTCL00 ) Ena Pur
Prf(DFHCICST) Uda Bel Iso
Tra(AREP) Pri( 001 ) Pro(DFH$AREP) Tc1( DFHTCL00 ) Ena Pur
Prf(DFHCICST) Uda Bel Iso
+ Tra(AUPD) Pri( 001 ) Pro(DFH$AALL) Tc1( DFHTCL00 ) Ena Pur
Prf(DFHCICST) Uda Bel Iso

```

Figure 158. 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 159.

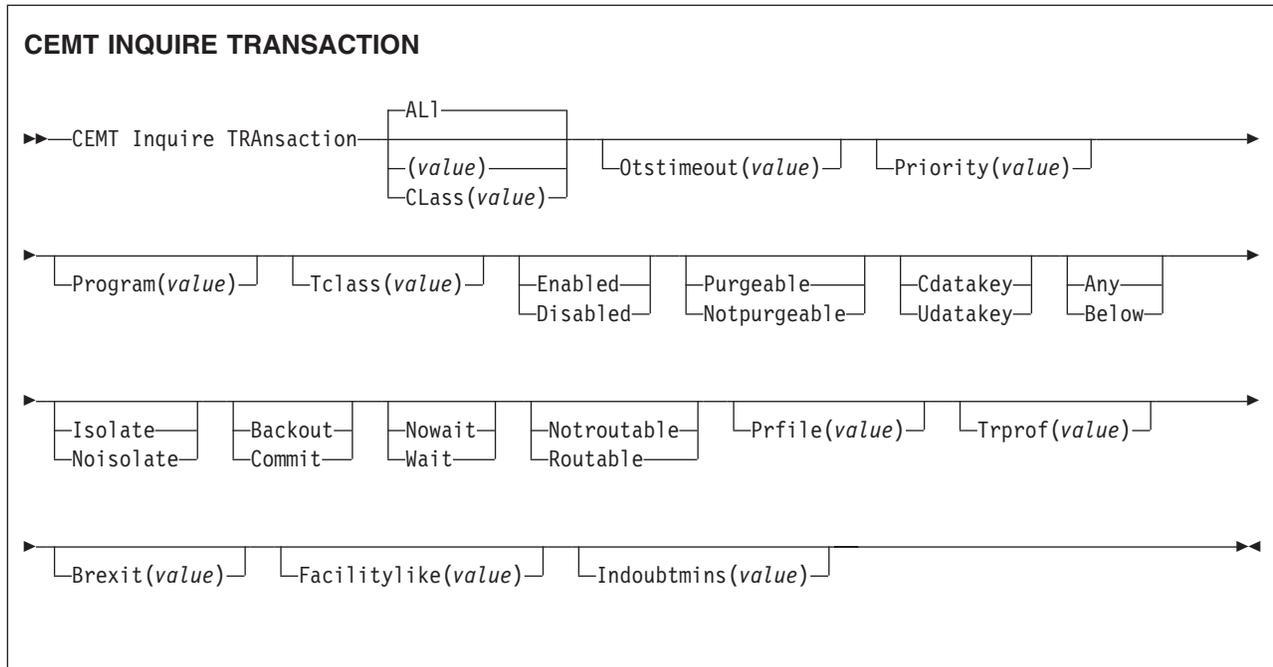
```

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)
Isolate(Isolate)
Trprof()
Indoubt
INDoubtwait
Brexid
Facilitylike
Otsttimeout(12000)

```

Figure 159. 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.

## CEMT INQUIRE TRANSACTION

### 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*.

### **Taskdata1oc**

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

**Function**

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 `cent 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:

- Overtyping your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 140).
- Use the CEMT SET TRDUMPCODE command.

(*value*)

is a 4-byte transaction dump code. See the *CICS Messages and Codes* manual for information on transaction dump codes.

ALI

is the default.

**Sample screen**

```
IN TRDU
STATUS: RESULTS - OVERTYPE TO MODIFY
Trd(ATNI) Tra          Loc Max( 999 ) Cur(0002)
Trd(ERRORS) Tra       Loc Max( 999 ) Cur(0001)
```

Figure 160. 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

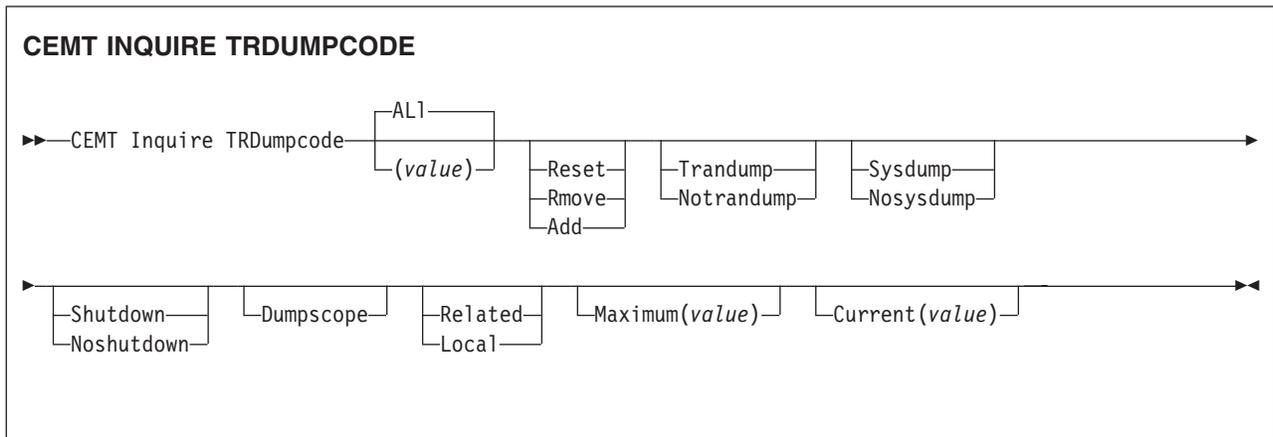
convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 161.

```

IN TRDU
RESULT - OVERTYPE TO MODIFY
  Trdumpcode(ATNI)
  Trandumping( Trandump )
  Sysdumping( Nosysdump )
  Shutoption( Noshutdown )
  Dumpscope( Local )
  Maximum( 999 )
  Current(0002)
  Action(           )
  
```

Figure 161. The expanded display of an individual entry



**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

## CEMT INQUIRE TRDUMPCODE

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 the *CICS Messages and Codes* manual for information on transaction dump codes.

## CEMT INQUIRE TSMODEL

### Function

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, overwrite your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 140).

### ALI

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 162. CEMT INQUIRE TSMODEL screen

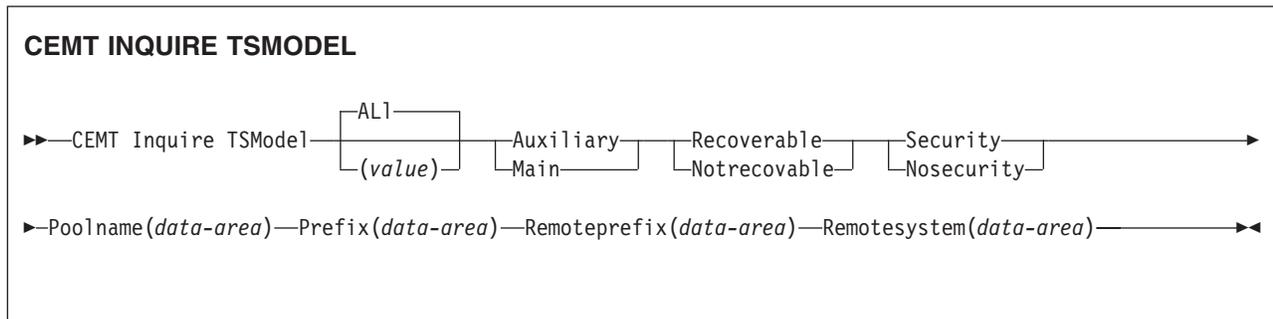
If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 163 on page 348.

## CEMT INQUIRE TSMODEL

```

I TSM
RESULT
  Tsmode1 (DFHWEB)
  Prefix (DFHWEB)
  Location (Main)
  Recovstatus (Notrecoverable)
  Securityst (Nosecurity)
  Poolname ()
  Remotesystem ()
  Remoteprefix ()
  
```

Figure 163. The expanded display of an individual entry



### 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.

**Notrecoverable** 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.

**Tsmode1** (*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

### Function

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 `cent 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, overwrite your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 140).

### ALI

is the default.

### (value)

is the name of the temporary storage pool for which information is requested.

### Sample screen

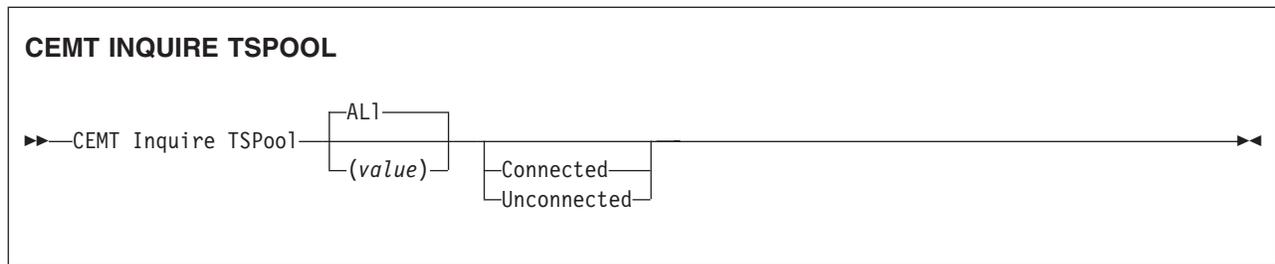
```
I TSP
STATUS: RESULTS
Tsp(DFHWEB ) Unc
```

Figure 164. 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 165.

```
I TSP
RESULT
Tspool (DFHWEB)
Connstatus (Unconnected)
```

Figure 165. The expanded display of an individual entry



## 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

**Function**

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, 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:

<b>**</b>	BMS paging
<b>\$\$</b>	BMS route
<b>X'fa' to X'ff'</b>	CICS
<b>CEBR</b>	Default CEBR queue name
<b>DF</b>	CICS
<b>DFHM</b>	Message cache for message-protected tasks
<b>DFxxxx</b>	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, overwrite your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 140).

**AL1**

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(AXBYQUEUEENAME1 ) Num(00003) Len(0000000192) Mai
Tra(CECI) Max(00064) Min(00064) Las( 00001267 )
Tsq(RDOQUEUEENAME4 ) Num(00001) Len(0000000064) Aux
Tra(CECI) Rec Max(00064) Min(00064) Las( 00000097 )
Tsq(RDOQUEUEENAME5 ) Num(00002) Len(0000000128) Aux
Tra(CECI) Rec Max(00064) Min(00064) Las( 00000086 )
Tsq(RDQUEUEENAME5 ) Num(00003) Len(0000000192) Mai
Tra(CECI) Max(00064) Min(00064) Las( 00000076 )

```

Figure 166. 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 167.

```

INQ TSQ
RESULT - OVERTYPE TO MODIFY
Tsqqueue(AXBYQUEUEENAME1)
Numitems(00003)
Length(0000000192)
Location(Main)
Action(          )
Transid(CECI)
Recovstatus(Notrecovable)
Maxitemlen(00064)
Minitemlen(00064)
Lastusedint(00001357)

```

Figure 167. 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.

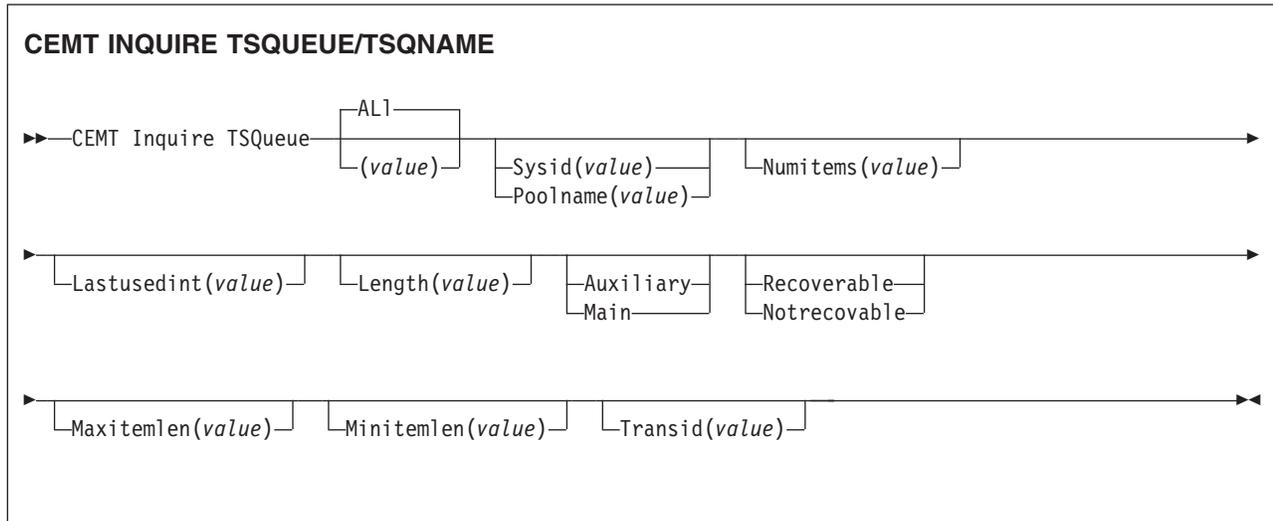
Such queues are browsable if there is a tsmodel or TST entry mapping the queue name to the TSPool.

**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 INQUIRE TSQUEUE / TSQNAME

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.



### 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. For information about how CICS calculates the length of items, see the *CICS System Programming Reference*.

#### **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. For information about how CICS calculates the length of items, see the *CICS System Programming Reference*.

#### **Minitemlen (value)**

displays the length in bytes of the smallest item in the temporary storage queue. For information about how CICS calculates the length of items, see the *CICS System Programming Reference*.

#### **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.

**Notrecoverable**

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

#### Function

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.

AL1

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 168. 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 169.

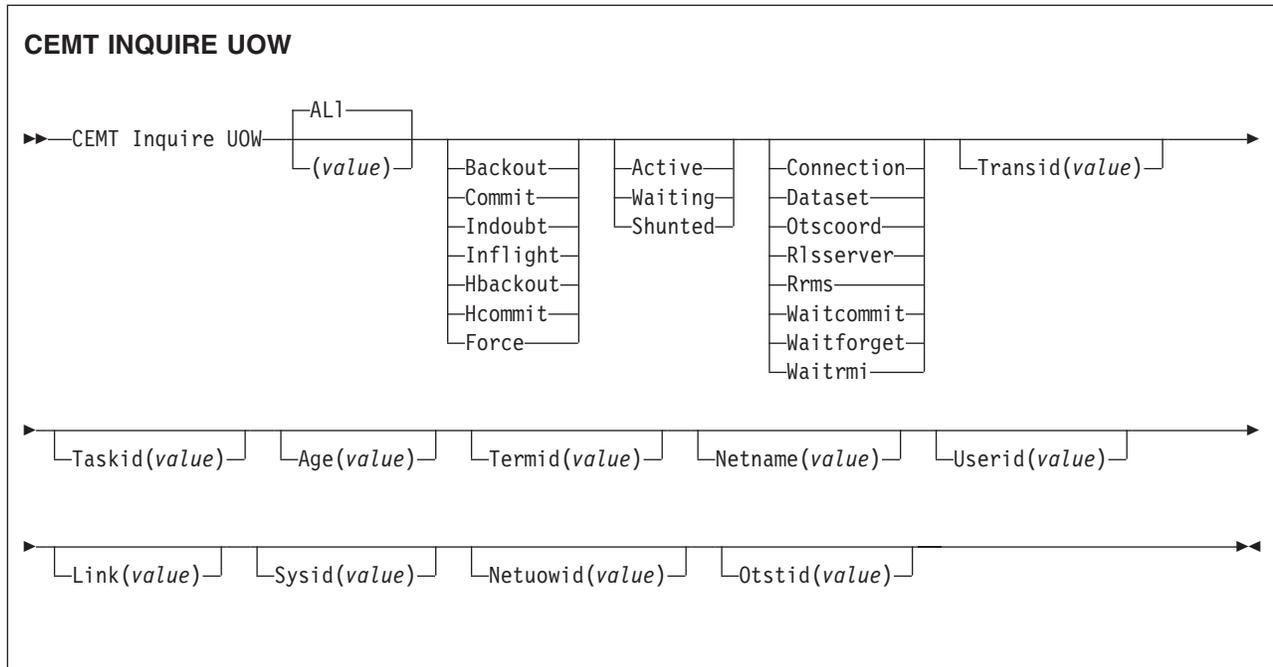
```

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 169. 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 or 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 UOWs 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:

<b>Backout</b>	This UOW is being backed out, or has failed to back out one or more of the recoverable resources involved in the UOW.
<b>Commit</b>	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.
<b>Indoubt</b>	This UOW is in the in-doubt state.
<b>Inflight</b>	The UOW is running normally.
<b>Hbackout</b>	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.
<b>Hcommit</b>	The UOW has been forcibly committed.

**Note:** You can reset this value by overtyping it with:

<b>Commit</b>	Attempt to force syncpoint commit processing, as specified for this UOW.
<b>Backout</b>	Attempt to force syncpoint backout processing, as specified for this UOW.
<b>Force</b>	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*.

## CEMT INQUIRE UOW

### **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:

<b>Connection</b>	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.
<b>Dataset</b>	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.
<b>Otscooord</b>	This UOW is waiting or was shunted due to the failure of the OTS coordinator.
<b>Rlsserver</b>	This UOW is waiting or was shunted for the OTS coordinator due to the failure of an RLS server.
<b>Rrms</b>	This UOW is waiting or has been shunted because communication has been lost with RRS/MVS.
<b>Waitcommit</b>	This UOW is waiting or was shunted because a failure occurred during commit processing.
<b>Waitforget</b>	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.
<b>Waitrmi</b>	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:

<b>Active</b>	The UOW is running normally, or is being unshunted.
<b>Waiting</b>	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.
<b>Shunted</b>	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

### Function

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(app1b.*)`, 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 170. 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

## CEMT INQUIRE UOWDSNFAIL

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 171.

```

INQ UOWDSN
RESULT
  Uowdsnfail
  Dsname(RLS.ACCOUNTS.ESDS.DBASE1)
  Cause(Dataset)
  Reason(Delexiterror)
  Uow(AA6DB080C40CEE01)
  Sysid()
  Netname()
  Rlsaccess(Rls)
  
```

Figure 171. The expanded display of an individual entry

**CEMT INQUIRE UOWDSNFAIL**

▶▶ CEMT Inquire UOWDsnfail—Uow(value)—Dsname(value) —————▶

Rls  
Notrls

Cache  
Connection  
Dataset  
Rlserver  
Undefined

▶————▶

Backupnonbwo  
Commitfail  
Datasetfull  
Deadlock  
Delexiterror  
Failedbkout  
Indexrecfull  
Indoubt  
Ioerror  
lckstrucfull  
notapplic  
Openererror  
Rlsgone  
Rrcommitfail  
Rrindoubt

Sysid(value)

Netname(value)

▶————▶

### 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.

**Rlsserver** 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 the section on moving data sets in 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 the section on moving data sets in 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

## CEMT INQUIRE UOWDSNFAIL

	<p>data set, and has now failed in-doubt. The connection to the coordinating system needs to be reestablished.</p>
<b>Ioerror</b>	<p>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.</p>
<b>Lckstrucfull</b>	<p>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.</p>
<b>Notapplic</b>	<p>The value for CAUSE is not CONNECTION, RLSSERVER, or DATASET.</p>
<b>Openererror</b>	<p>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.</p>
<b>Rlsgone</b>	<p>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.</p>
<b>Rrcommitfail</b>	<p>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.</p>
<b>Rrindoubt</b>	<p>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.</p>

Each Reason corresponds to only one Cause value, as shown in the following table:

<b>Cause</b>	<b>Reason</b>
CACHE	NOTAPPLIC
CONNECTION	INDOUBT
CONNECTION	RRINDOUBT
DATASET	BACKUPNONBWO
DATASET	DELEXITERROR

<b>Cause</b>	<b>Reason</b>
DATASET	DATASETFULL
DATASET	DEADLOCK
DATASET	FAILEDDBKOUT
DATASET	INDEXRECFULL
DATASET	IOERROR
DATASET	LCKSTRUCFULL
DATASET	OPENERROR
RLSSERVER	COMMITFAIL
RLSSERVER	RLSGONE
RLSSERVER	RRCOMMITFAIL
UNDEFINED	Not applicable

**R1saccess**

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

### Function

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.

### Notes:

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 in the *CICS System Programming Reference* manual.)
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 cent 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          ) R1e(015) Enq(00000000)
Uow(AC143001D4A72601) Tra(CECI) Tas(0000038) Act Add Own
  Res(g...                      ) R1e(004) Enq(00000000)

```

Figure 172. 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 overwrite 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 173.

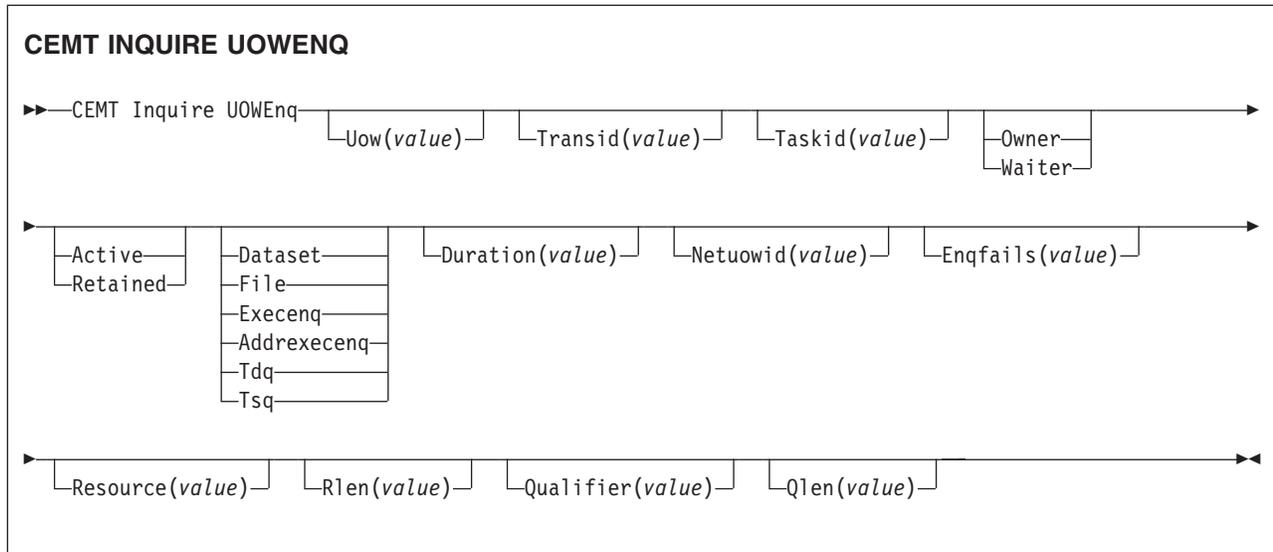
```

INQ UOWE
RESULT
Uowenq
Uow(AC143001D4A72601)
Transid(CECI)
Taskid(0000038)
State(Active)
Type(Execenq)
Relation(Owner)
Resource(ACCOUNTS.DBASE1)
R1en(015)
Enqfails(00000000)
Netuowid(..GBIBMIYA.IYA2T5C2...Mx...)
Qualifier()
Q1en(000)

```

Figure 173. 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 370. 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 370.

**Note:** Nondisplayable characters appear as periods. You can use PF2 on the expanded panel: to see the value in hexadecimal.

**R1en (*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.
- Addrxecenq** 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.

## CEMT INQUIRE UOWENQ

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
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

### Function

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)*

is an 8-byte token identifying the UOW-link for which data is to be returned.

AL1

is the default.

### Sample screen

```
I UOWL
STATUS: RESULTS - OVERTYPE TO MODIFY
Uow1(01840009) Uow(AC142A977E6F3204) Con Lin(IYLX1  )
      Coo Appc War Sys(ISC1)           Net(..GBIBMIYA.IYCWTC13..p...
Uow1(01C40006) Uow(AC142A977E6F3204) Con Lin(IYLX3  )
      Sub Irc  War Sys(MR03)           Net(..GBIBMIYA.IYCWTC13..p...
```

Figure 174. 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 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).

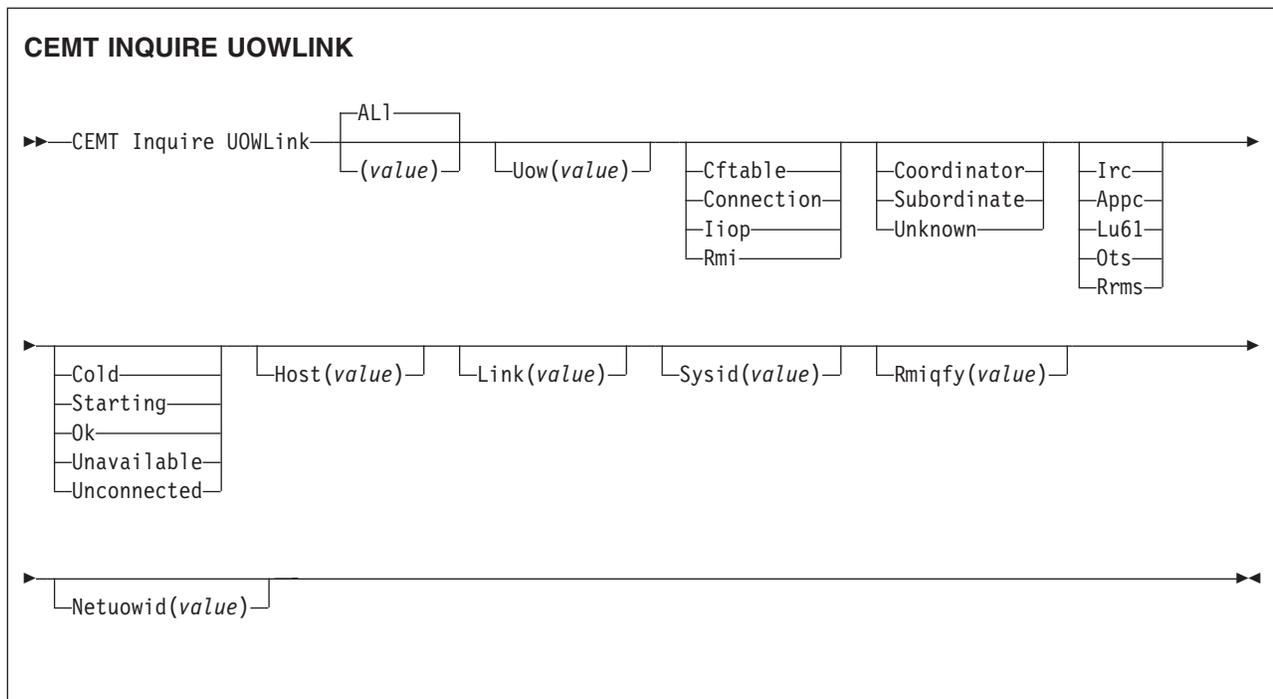
If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 175 on page 372.

## CEMT INQUIRE UOWLINK

```

I UOWL
RESULT - OVERTYPE TO MODIFY
Uowlink(01840009)
Uow(AC142A977E6F3204)
Type(Connection)
Host( )
Link(IYLX1)
Action( )
Role(Coordinator)
Protocol( )
Resyncstatus( )
Sysid( )
Rmiqfy( )
Netuowid( )
Urid
    
```

Figure 175. The expanded display of an individual entry



### Displayed fields

#### Host (value)

displays a 255-character area into which is returned, for TYPE of IIOP, 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. For TYPE of CONNECTION, CFTABLE, or RMI, HOST displays blanks. This is useful for identifying the participant especially when problems occur. Notes that the UOW and the participant may belong to the same CorbaServer.

#### Link (value)

displays, for a TYPE value of CONNECTION, the 8-character netname of the remote system. For a TYPE value of RMI, LINK displays the entry name of the task-related user exit. For a TYPE value of CFTABLE, LINK displays the

8-character name of the coupling facility data table pool. For a type value of IIOp, LINK displays blanks; the equivalent information for IIOp UOWLINKs is displayed by the HOST attribute.

**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:

<b>Irc</b>	Inter-Region Communication. This is an MRO connection.
<b>Appc</b>	Advanced Program to Program Communication.
<b>Lu61</b>	LUTYPE 6.1.
<b>Ots</b>	Object Transaction Services for Java programs.
<b>Rrms</b>	indicates that the UOW is coordinated by RRS/MVS.

**Resyncstatus**

displays the resynchronization status of the connection. The values are:

<b>Cold</b>	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.
<b>Starting</b>	The connection is being acquired, but the exchange lognames process has not yet completed.
<b>Ok</b>	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.
<b>Unavailable</b>	The connection is not currently acquired.
<b>Unconnected</b>	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 a connection type of CONNECTION, CFTABLE, or IIOp, RMIQFY displays blanks.

**Role**

displays the role of the connection. The values are:

<b>Coordinator</b>	This connection is to the syncpoint coordinator for the UOW.
<b>Subordinate</b>	This connection is to a syncpoint subordinate for the UOW.
<b>Unknown</b>	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, CFTABLE or IIOp, or the PROTOCOL field is showing RRMS, SYSID displays blanks.

**Type**

displays the type of connection. The values are:

<b>Cftable</b>	A connection to a CFDT server.
<b>Connection</b>	A connection defined by a CONNECTION resource definition.
<b>liop</b>	An OTS link.
<b>Rmi</b>	A connection to an external resource manager using the resource manager interface (RMI).

## CEMT INQUIRE UOWLINK

**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.

## CEMT INQUIRE URIMAP

### Function

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 TCPIP SERVICE 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 140).

### ALI

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 176. 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 177 on page 376.



**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 TCPIP SERVICE 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 TCPIP SERVICE definition that specifies the inbound port to which this URIMAP definition relates. If this definition is not given, the URIMAP definition relates to all TCPIP SERVICE 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.

## CEMT INQUIRE URIMAP

| **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 HFS 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.

| **Characteraset**(*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

### Function

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:

- Overtyping your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 140).
- Use the CEMT SET VTAM command.

### Sample screen

```

IN VT
STATUS: RESULTS - OVERTYPE TO MODIFY
Vta Ope Psd( 000000 )
    
```

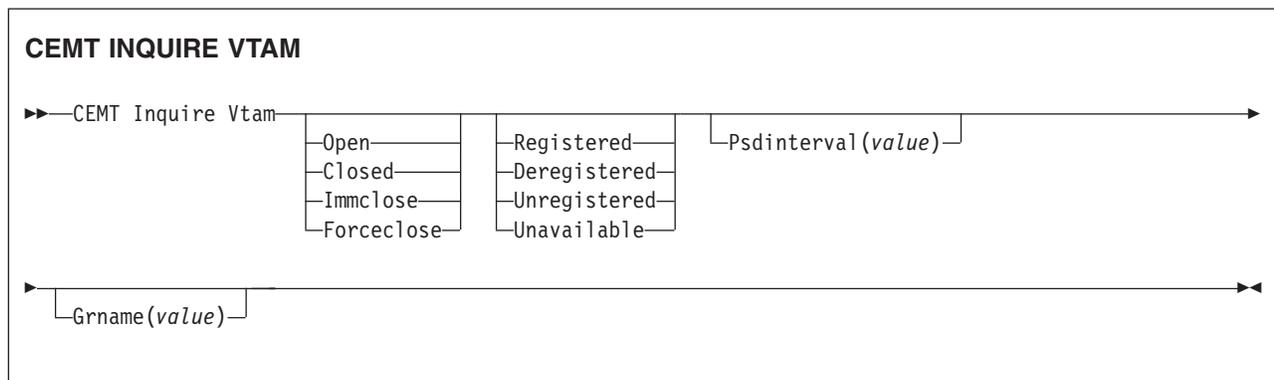
Figure 178. CEMT INQUIRE VTAM screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 179.

```

IN VT
RESULT - OVERTYPE TO MODIFY
Vtam
Openstatus( Open )
Psdinterval( 000000 )
Grstatus()
Grname()
    
```

Figure 179. The expanded display



## 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 an 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

### Function

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:

- Overtyping your changes on the INQUIRE screen after tabbing to the appropriate field (see “Overtyping a display” on page 140).
- Use the CEMT SET WEB command.

### Sample screen

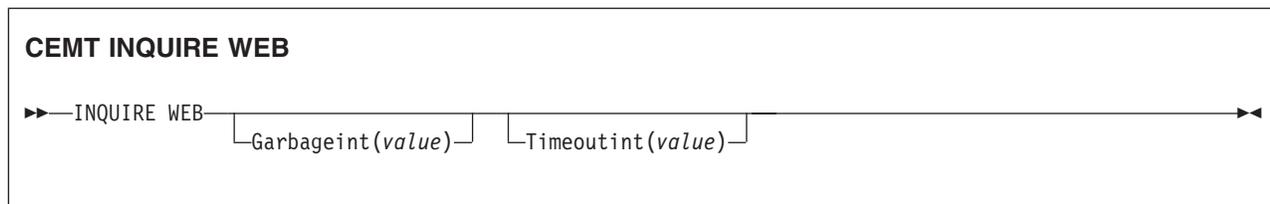
```
IN WE
STATUS: RESULTS - OVERTYPE TO MODIFY
Web Gar( 00060 ) Tim( 00005 ) Loc
```

Figure 180. CEMT INQUIRE WEB screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 179 on page 380.

```
I WEB
RESULT - OVERTYPE TO MODIFY
Web
Garbageint( 00060 )
Timeoutint( 00005 )
```

Figure 181. The expanded display



### 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

### Function

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 W). You get a screen that lists the current status.
- Type CEMT INQUIRE WEBSERVICE (CEMT I W) 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 W en`, the resulting display will show you the details of only those process-types that are enabled.

To change various attributes, you can:

- Overtyping your changes on the INQUIRE screen after tabbing to the appropriate field.
- Use the CEMT SET WEBSERVICE command.

### ALI

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.

Sample screen

```

I WEBSERVICE
STATUS: RESULTS - OVERTYPE TO MODIFY
Webs(INQUIRE1          ) Pip(SAMPLE  )
   Ins                Pro(DFH0XCMN) Com                Dat(20050119)
Webs(INQUIRE2          ) Pip(SAMPLE  )
   Ins                Pro(DFH0XCMN) Com                Dat(20050119)
Webs(PLACE1            ) Pip(SAMPLE  )
   Ins                Pro(DFH0XCMN) Com                Dat(20050119)
Webs(WEBSESV2          ) Pip(LID852  )
   Ins                Pro(WEBSESV2) Cha Con(DFHWS-DATA  ) Dat(20050120)

                                     SYSID=CIAP APPLID=IYK2ZAF1
RESPONSE: NORMAL                      TIME: 09.15.41 DATE: 09.02.05
PF 1 HELP          3 END          5 VAR          7 SBH 8 SFH 9 MSG 10 SB 11 SF
    
```

Figure 182. CEMT INQUIRE WEBSERVICE screen

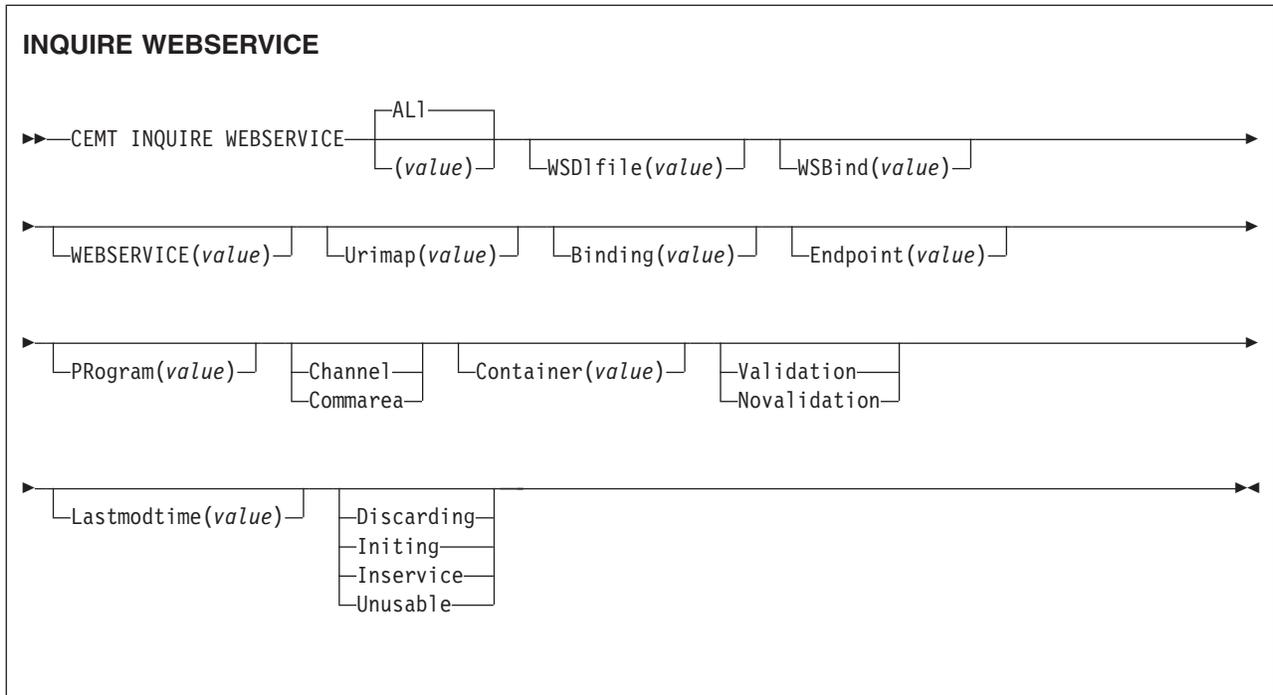
If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 183.

```

I WEBSERVICE(INQUIRE1)
RESULT - OVERTYPE TO MODIFY
  Webservice(INQUIRE1)
  Pipeline(SAMPLE)
  Validationst( Novalidation )
  State(Inservice)
  Urimap()
  Program(DFH0XCMN)
  Pgminterface(Commarea)
  Container()
  Datestamp(20050119)
  Timestamp(10:47:42)
  Wsdlfile(/u/chrisb/wsdl/beta4/inquireSingle.wsdl)
  Wsbind(/u/dbear1/sampbind/inquireSingle.wsbind)
  Endpoint(http://my-server:my-port/exampleApp/inquireSingle)
  Binding(DFH0XCMNHTTPSBinding)

                                     SYSID=CIAP APPLID=IYK2ZAF1
                                     TIME: 09.16.38 DATE: 09.02.05
PF 1 HELP 2 HEX 3 END          5 VAR          7 SBH 8 SFH          10 SB 11 SF
    
```

Figure 183. The expanded display of an individual entry



**Conditions:** NOTAUTH, NOTFND

### Displayed fields

#### **WSD1file**(value)

Displays the name of the Web service description file associated with the WEBSERVICE resource. The name can be up to 255 characters long.

#### **WSBind**(value)

Displays the name of the Web service binding file. The name can be up to 255 characters long.

#### **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.

#### **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.

#### **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.

#### **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.

## CEMT INQUIRE WEBSERVICE

### **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.

### **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

### **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.

### **Validationst**

Indicates whether full validation of SOAP messages is currently enabled for this WEBSERVICE:

#### **Validation**

Full validation is enabled

#### **Disabled**

Full validation is disabled

### **Lastmodtime**(*value*)

Displays the time that the deployed WSBind file on HFS 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 HFS 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 HFS file pointed to by the WEBSERVICE definition, at the time the WEBSERVICE was installed.

If you issue an INQUIRE WEBSERVICE command before a newly-installed or updated WEBSERVICE has fully initialized, the returned LASTMODTIME value will be zero.

## CEMT INQUIRE WORKREQUEST

### Function

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 IIOF 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 IIOF.

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.

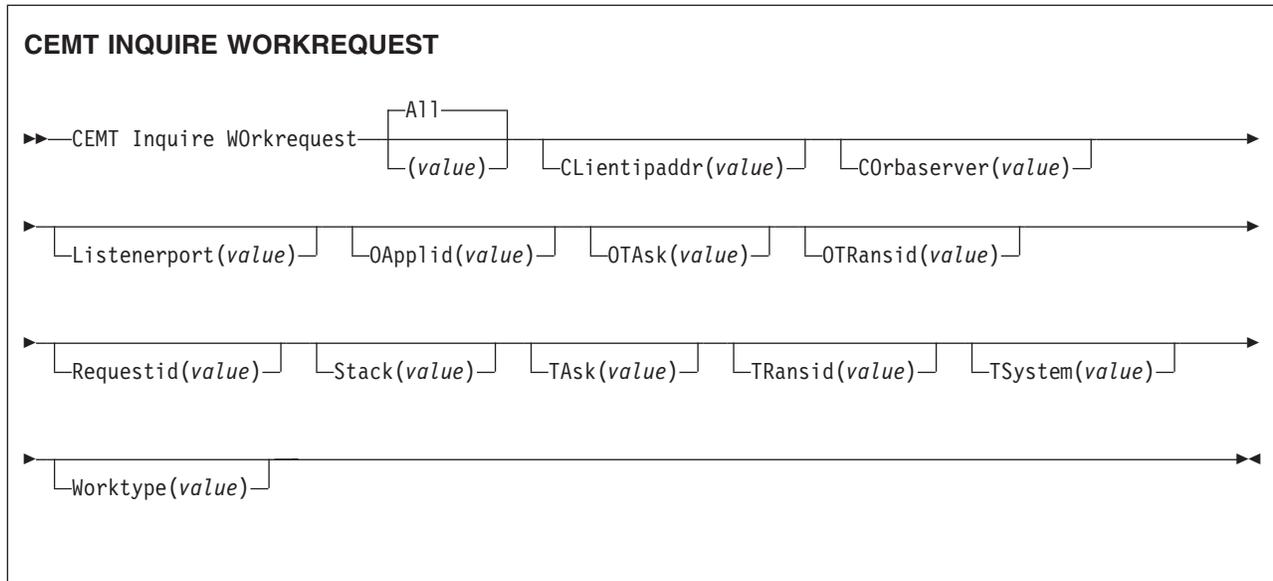
### Input

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 W0). You get a display that lists the current status.
- Type CEMT INQUIRE WORKREQUEST (CEMT I W0) 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 `cent 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.

#  
#  
#  
#  
#

**A11**

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(IYK2ZJV1) Ota(0000101)
Wor(15BED280) Iio Tas(0000090) Tra(CIRP)
  Cli(9.20.7.123      ) Lis(20142) Cor(CHN1) Oap(IYK2ZJV1) Ota(0000101)
Wor(15BED6E0) Iio Tas(0000176) Tra(CIRP)
  Cli(9.20.7.123      ) Lis(20142) Cor(CHN1) Oap(IYK2ZJV1) Ota(0000251)
Wor(15BEDBC0) Iio Tas(0000176) Tra(CIRP)
  Cli(9.20.7.123      ) Lis(20142) Cor(CHN1) Oap(IYK2ZJV1) Ota(0000251)

                                     SYSID=JOHN APPLID=IYK2ZJV1
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 184. 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 185.

```

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 185. The expanded display of an individual entry

## Displayed fields

### **Clientipaddr**(*value*)

displays the TCP/IP 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 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.

## CEMT INQUIRE WORKREQUEST

As each new request is received from the client the request receiver creates a stack and initialises 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 workrequest 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 TCP/IP 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 TCP/IP. 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 DELETSHIPED)
- 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

### Function

You can use the PERFORM CLASSCACHE command to initialize (start or reload) and terminate (phase out, purge or forcepurge) the shared class cache. While you are performing one of these operations, you can use other options on the command to set attributes of the shared class cache, as follows:

- When you are **initializing** the shared class cache (PERFORM CLASSCACHE Start or Reload), you can change its size (Cachesize option), and change the JVM profile that is used for the master JVM (Profile option).
- When you are **terminating** the shared class cache (PERFORM CLASSCACHE Phaseout, Purge or Forcepurge), 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.

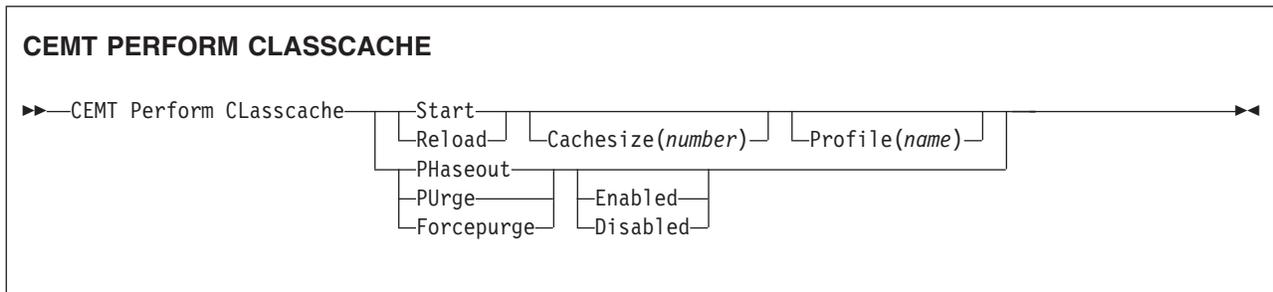
### Description

The PERFORM CLASSCACHE command is intended to operate on the active shared class cache in the CICS region. The Start and Reload operations only affect this shared class cache. The Phaseout, Purge and Forcepurge operations, however, also act on any old shared class caches that are still present in the region because they are waiting for worker JVMs that are dependent on them to be phased out. For example, if you issue a CEMT PERFORM CLASSCACHE Forcepurge command, all tasks using worker JVMs dependent on a shared class cache are forcepurged, both those dependent on the current shared class cache and those dependent on old shared class caches. Note, though, that JVMs running independently of the shared class cache are *not* affected by the CEMT PERFORM CLASSCACHE Phaseout, Purge and Forcepurge commands.

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.

When you use the Cachesize, Profile and Autostartst options, the new settings override the size, JVM profile and autostart status that were originally set by the system initialization parameters JVMCCSIZE, JVMCCPROFILE and JVMCCSTART. Subsequent CICS restarts use the most recent settings that you made using the PERFORM CLASSCACHE command (or the SET CLASSCACHE command), unless the system is INITIAL or COLD started, or the system initialization parameters are specified as overrides at startup. In these cases, the settings from the system initialization parameters are used.

## Syntax



## Options

The operations you can perform with the PERFORM CLASSCACHE command are:

### Forcepurge

stops and deletes the shared class cache. All tasks using worker JVMs dependent on the shared class cache are terminated by the SET TASK FORCEPURGE mechanism, and the worker JVMs are terminated. No more JVMs can use the shared class cache, and it is deleted when all the worker JVMs that were dependent on it have been terminated.

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.

### Phaseout

stops and deletes the shared class cache. All worker JVMs using the shared class cache are marked for deletion. The JVMs are actually deleted when they finish running their current Java programs. No more JVMs can use the shared class cache, and it is deleted when all the worker JVMs that were dependent on it have been terminated.

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.

### Purge

stops and deletes the shared class cache. All tasks using worker JVMs dependent on the shared class cache are terminated by the SET TASK PURGE mechanism, and the worker JVMs are terminated. No more JVMs can use the shared class cache, and it is deleted when all the worker JVMs that were dependent on it have been terminated.

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

## CEMT PERFORM CLASSCACHE

a JVM whose profile requires the use of the shared class cache will fail. Specify the `Autostartst` option to set the status of autostart.

### 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 `Profile` options are used in the creation of the cache. If newer versions of Java classes are available to be loaded, the new class cache uses them. However, worker JVMs, both those that are already allocated to tasks and those that are allocated to tasks after you issue the command, continue to use the existing shared class cache and the old versions of the Java classes until the new shared class cache is ready. When the new shared class cache is ready, subsequent requests for worker JVMs are given a worker JVM that uses the new cache. These new worker JVMs are started as they are requested by applications, and they replace the worker JVMs that are using the old shared class cache. The worker JVMs that are using the old shared class cache are allowed to finish running their current Java programs, and then they are terminated. The old shared class cache is deleted when all the worker JVMs that are dependent on it have been terminated.

### 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 `Profile` options are used in the creation of 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*)

If you are performing the `Start` operation, you can use `Cachesize` to specify the size of the shared class cache that is to be started. If you are performing the `Reload` operation, you can use `Cachesize` to specify the size of the new shared class cache that is to be loaded. 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.

### **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, an explicit `CEMT PERFORM CLASSCACHE Start` command (or the equivalent `EXEC CICS` command) is required to start the shared class cache. If the status of the shared class cache is `STOPPED` and autostart is disabled, and `CICS` receives a request to run a Java application in a JVM whose profile requires the use of the shared class cache, the request fails.

### **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, 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*)

If you are performing the `Start` operation, you can use `Profile` to specify the eight-character name of the JVM profile that will be used for the master JVM

that starts the shared class cache. If you are performing the Reload operation, you can use Profile to specify the eight-character name of the JVM profile that will be used for the master JVM that loads the new shared class cache.

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

### Function

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.

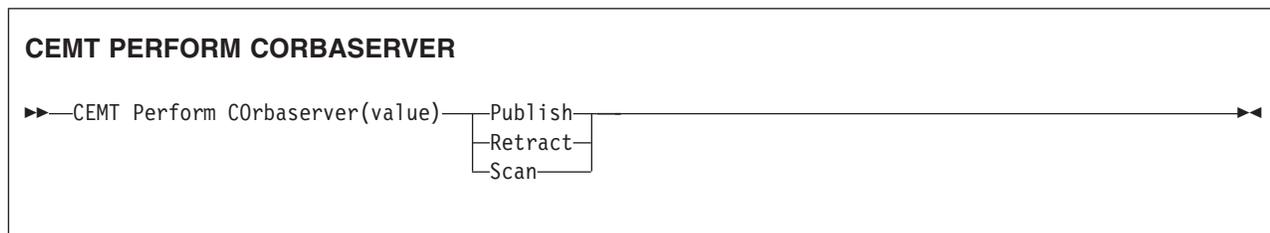
You can issue a PERFORM CORBASERVER command only when the CorbaServer is in a steady state—that is, when it is in ENABLED or DISABLED state, but *not* when it is in ENABLING, DISABLING, or DISCARDING state.

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 the hierarchical file store (HFS) in order to launch a JVM. This requires that the CICS region userid is authorized to z/OS UNIX System Services to enable CICS to access the required HFS directories, such as those specified on the JVMPROPS, LIBPATH, and CLASSPATH parameters.

### Description

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



### 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 DELETSHIPED

## CEMT PERFORM DELETSHIPED

### Function

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 DELETSHIPED IDLE command. The interval between scheduled invocations of the mechanism is specified on the DSHIPINT system initialization parameter, or on a subsequent SET DELETSHIPED INTERVAL command.

PERFORM DELETSHIPED invokes timeout delete immediately.

### Notes:

1. Shipped definitions are not deleted if there is an automatic initiate descriptor (AID) associated with the terminal.
2. Issuing a PERFORM DELETSHIPED 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 DELETSHIPED

▶▶—CEMT Perform DEletshipped—▶▶

## CEMT PERFORM DJAR

### Function

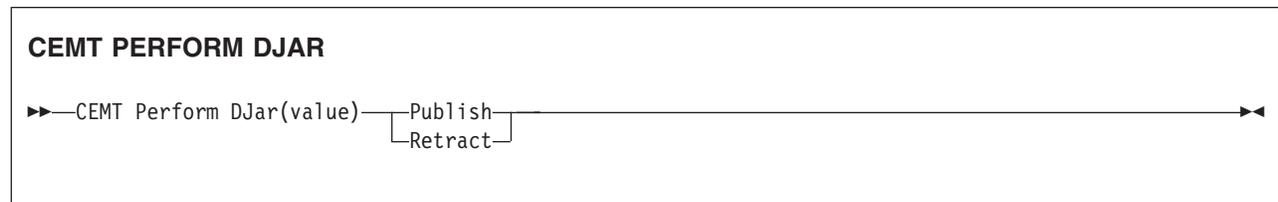
The PERFORM DJAR command performs certain actions against a specified deployed JAR file. If no action is specified, the command has no effect.

In addition to the traditional DFSMS-managed data sets, such as partitioned data sets (PDS), CICS requires access to z/OS UNIX System Services and the hierarchical file store (HFS) in order to launch a JVM. This requires that the CICS region userid is authorized to z/OS UNIX System Services to enable CICS to access the required HFS directories, such as those specified on the JVMPROPS, LIBPATH, and CLASSPATH parameters.

### Description

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



### 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 ENDAFFINITY

### Function

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

```

▶▶—CEMT Perform Endaffinity—NETName(value)—┐
└──────────────────────────────────┘
└──────────────────┘

```

### 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 PIPELINE

### Function

Use the PERFORM PIPELINE command to initiate a scan of the Web service binding directory that is specified in the WSBIND attribute of the PIPELINE definition. If the WSBIND attribute is not specified, there is nothing to scan, and control returns to your program.

If the directory location specified is valid, CICS examines the Web service binding files in the directory to determine if they should be installed into the system:

- CICS installs any files it finds that have not been installed already.
- If a file has been installed already, but the file in the directory is newer than the one currently in use, the one that is in use is discarded, and the newer file is installed in its place.

If, for any reason, CICS fails to install an individual Web service binding file, processing continues with the remaining files in the directory. When the scan completes, the PIPELINE is available for use with whichever of the binding files were installed successfully.

### Description

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

▶▶—CEMT Perform PIpeline(value)—Scan————▶▶

### Options

#### PIPELINE (*value*)

specifies the 8-character PIPELINE name.

#### Scan

Scan the PIPELINE's Web service binding directory

## CEMT PERFORM RESET

### Function

Synchronize the CICS date and time-of-day with the system date and time-of-day.

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 Reset—▶▶

## CEMT PERFORM SECURITY

## CEMT PERFORM SECURITY

### Function

Rebuild the in-store RACF resource profiles. The RACF resource profiles are rebuilt by reissuing the RACLIST macro for the active class names.

### Syntax



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

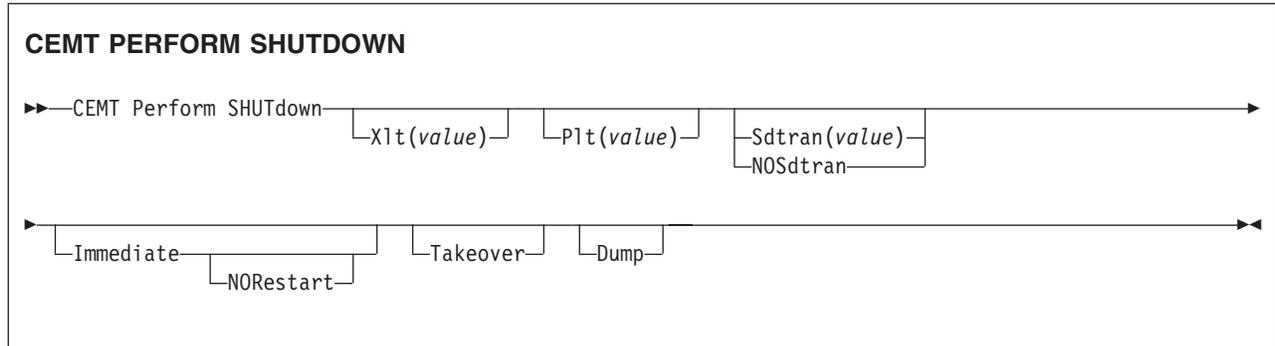
is an optional keyword. It does not alter the action of the transaction.

## CEMT PERFORM SHUTDOWN

### Function

# Shut down the CICS region. 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



### 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.

#### P1t (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.

#### X1t (value)

specifies the 2-character suffix of a transaction list table (XLT) to be used for the shutdown.

## CEMT PERFORM 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		
	<b><i>(normal)</i></b>	<b><i>TAKEOVER</i></b>	<b><i>IMMEDIATE</i></b>
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		
	<b><i>(normal)</i></b>	<b><i>TAKEOVER</i></b>	<b><i>IMMEDIATE</i></b>
CICS alternate	shuts down	takes over	takes over

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

## CEMT PERFORM STATISTICS

### Function

Write the statistics for a named resource type immediately to the SMF data set, rather than wait for the current statistics-gathering interval to expire. The recording is made irrespective of the ON or OFF setting of statistics. See also page “CEMT INQUIRE STATISTICS” on page 295 and page “CEMT SET STATISTICS” on page 483 for a description of the CEMT INQUIRE and SET STATISTICS commands.

### Description

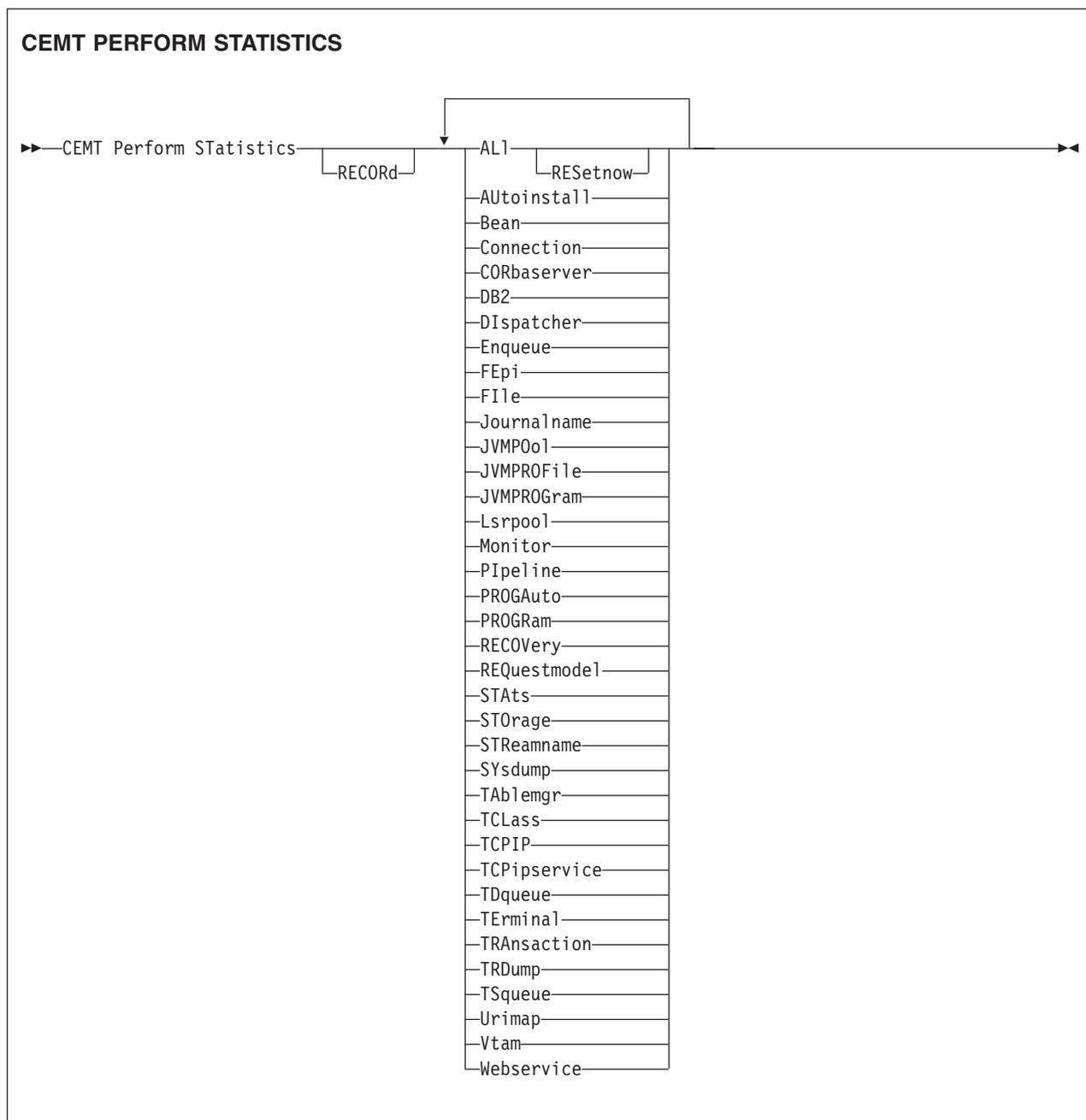
You can request either statistics for all resources (ALL), or statistics for individual resources in various combinations by selecting one or more keywords from the list.

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.

# CEMT PERFORM STATISTICS

## Syntax



## Options

**AL1** 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.

**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.

**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.

**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.

**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.

## CEMT PERFORM STATISTICS

### **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.

### **TErминаl**

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

## CEMT SET AUTOINSTALL

### Function

Change autoinstall values.

### Description

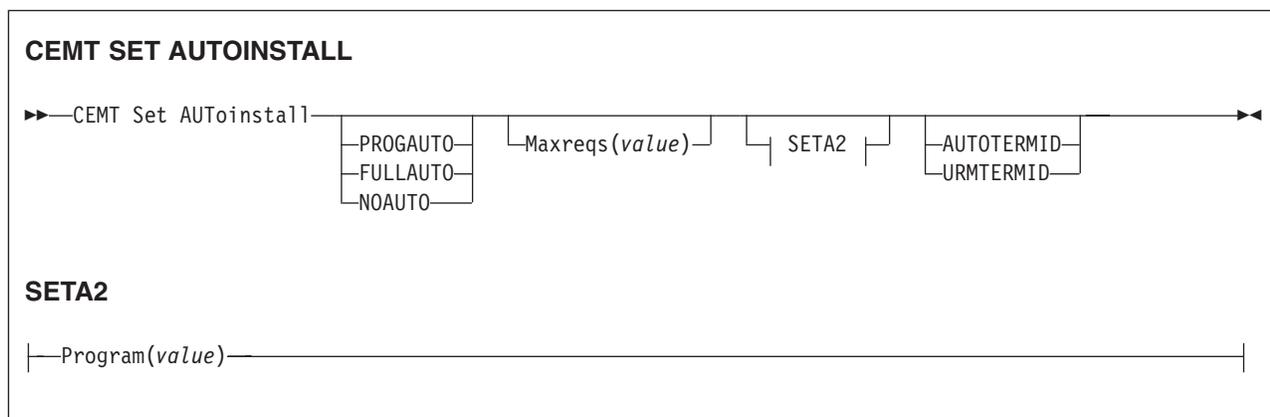
The CEMT SET AUTOINSTALL command allows you to change some of the values that control the automatic installation (autoinstall) of VTAM terminals, virtual terminals including bridge facilities used by the 3270 bridge mechanism, and MVS consoles. For information about autoinstall, see *CICS Resource Definition 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 overwrite 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.



### 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

## CEMT SET AUXTRACE

### Function

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 19, “CETR—trace control,” on page 533.

### 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 overwrite 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		
▶▶—CEMT Set AUXtrace		◀◀
—STArt—	—NOswitch—	—Switch—
—Pause—	—NExt—	
—STOp—	—All—	

### Options

#### A11

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

### Function

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 overwrite 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) re1` will release the specified bridge facility.

To obtain a syntax prompt, enter ?CEMT S BR.

```
CEMT SET BRFACILITY
▶▶—CEMT Set BRfacility—(value)————▶▶
                        [ALL]         [REleased]
```

### 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

### Function

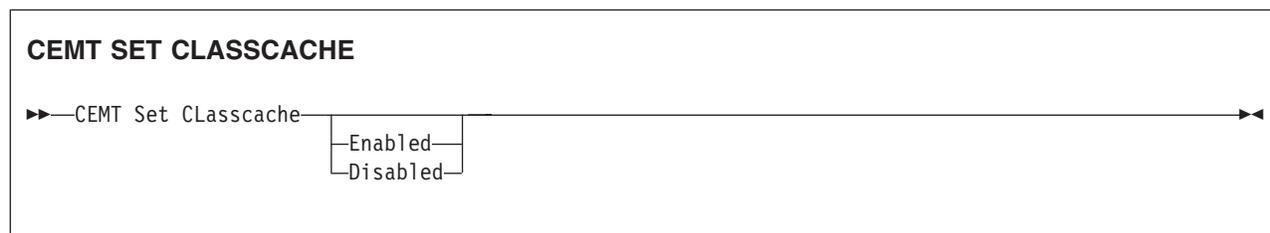
Set the status of autostart for the shared class cache.

### 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. You can change the status of autostart by using the SET CLASSCACHE command, or by using the Enabled or Disabled option on the CEMT PERFORM CLASSCACHE command (or the equivalent EXEC CICS command) when you are performing the Phaseout, Purge or Forcepurge operations with that command.

When you change the status of autostart for the shared class cache while CICS is running, subsequent CICS restarts use the most recent settings that you made using the CEMT or EXEC CICS SET CLASSCACHE command (or the CEMT or EXEC CICS PERFORM CLASSCACHE command), unless the system is INITIAL or COLD started, or the system initialization parameter JVMCCSTART is specified as an override at startup. In these cases, the setting from the system initialization parameter is used.

### Syntax



### 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 CEMT PERFORM CLASSCACHE Start command (or the equivalent EXEC CICS command) is required to start it. If the status of the shared class cache is STOPPED and autostart is disabled, and CICS receives a request to run a Java application in a JVM whose profile requires the use of the shared class cache, the request fails.

#### 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

### Function

Change the status of connections linked to specific terminals.

### Description

When a connection has been defined as INDIRECT, no information is given unless the direct connection to which it refers is installed.

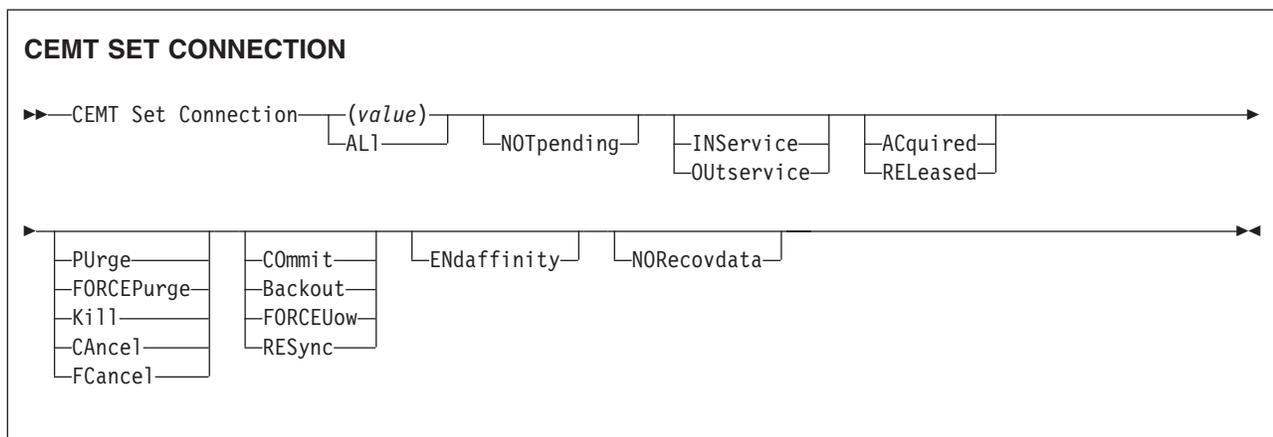
If you want to install a new connection definition when one is already installed, you must set the connection OUTSERVICE and, if it is VTAM-connected, release it before using the CEDA INSTALL commands for your new definition. See the *CICS Resource Definition Guide* for further information about connections.

### Syntax

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

- Type CEMT SET 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 overwrite 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 a l 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.



### 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.

**AL1**

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.

**CAnce1**

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.

## CEMT SET CONNECTION

### Notes:

1. There is no facility in VTAM for inquiring on affinities, so CICS has no certain knowledge that an affinity exists for a given connection. Whenever there is a possibility that an affinity has been created that 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 419 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 Transaction Server for z/OS partner that has performed an initial start

## CEMT SET CONNECTION

- A connection to a pre-CICS Transaction Server for z/OS 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 Transaction Server for z/OS partner, a lognames mismatch indicates that the partner has performed an initial start. For a pre-CICS Transaction Server for z/OS partner, a lognames mismatch indicates that the partner has performed a cold start. In either case, the recovery protocol has been corrupted by a loss of log data at the partner.

It is not possible to set a connection to NOTPENDING state (forcing in-doubt and erasing NOFORGET UOWs) until this system has made contact with the partner and received a new logname from it.

Decisions for a whole connection can be forgotten, but that does not affect the memory of a decision for any other connection involved in the UOW.

### Notes:

1. If you specify NOTPENDING you cannot specify COMMIT, BACKOUT, FORCEUOW, RESYNC, or NORECOVDATA. For advice on which option to use, see page “CEMT SET CONNECTION - notes on RES ync” on page 423.
2. NOTPENDING has no effect on MRO connections to pre-CICS Transaction Server for z/OS systems. However, the cold start of a pre-CICS Transaction Server for z/OS MRO partner causes the SET CONNECTION NOTPENDING function to be performed automatically, session by session.

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 Transaction Server for z/OS 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.

**Notes:**

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

### Function

Enable or disable a CorbaServer, turn its automatic publishing feature on or off, and set the session bean timeout value.

### Description

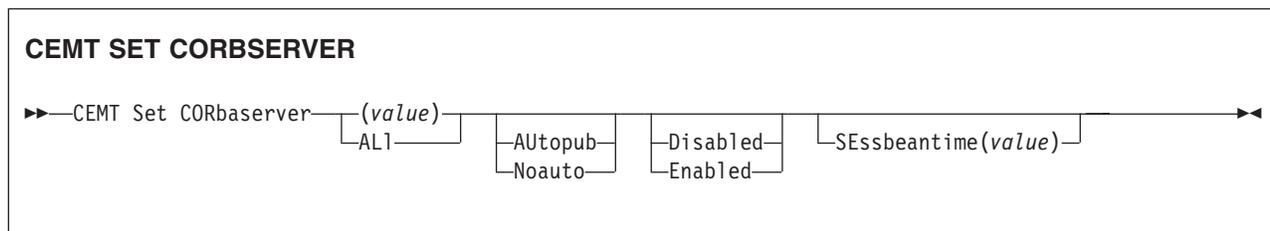
If the optional attributes are omitted, the command has no effect.

### 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 overwrite 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 a l 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.



### 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

## CEMT SET DB2CONN

### Function

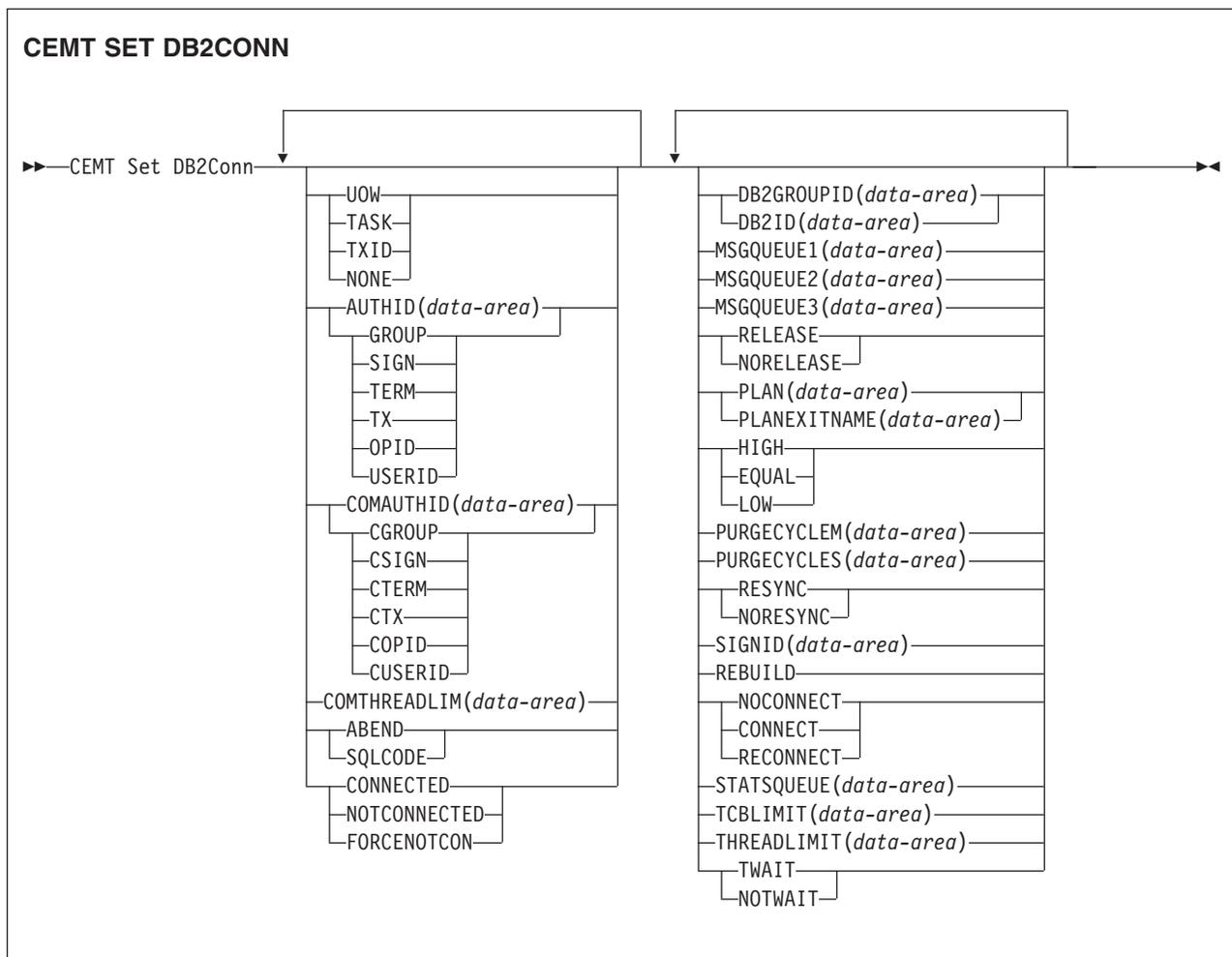
Set attributes of the CICS DB2 connection including the pool and command thread attributes.

### Syntax

Press the Clear key to clear the screen. To commence this transaction:

- Type CEMT SET DB2CONN (the minimum abbreviation is CEMT S DB2C). You get a display that lists the current status, similar to that obtained by CEMT INQUIRE DB2CONN. 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.



### Options

#### A ccountrec

specifies the minimum amount of DB2 accounting required for transactions

requiring pool threads. The specified minimum may be exceeded, as described in the following options. CVDA values are:

**UOW** The CICS DB2 attachment facility causes an accounting record to be produced 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 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.

Because pool threads are typically used by a number of different transaction IDs, there is an increased chance that a transaction containing multiple UOWs will use a different thread for each UOW (assuming the thread is released at syncpoint. In this case an accounting record may be produced for each UOW.

**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 should be used 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:

**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.

## CEMT SET DB2CONN

### 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 should be used 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:

### 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 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 to be passed back to an application that has issued a SQL request. CVDA values are:

**ABEND**

The application is abended with abend code AEY9.

**SQLCODE**

The application receives a -923 SQLCODE.

**Connectst**

sets the status of the CICS DB2 connection; that is, to start or to stop the CICS DB2 connection. CVDA values are:

**CONNECTED**

This is equivalent to issuing DSNB STRT to start the CICS DB2 attachment. If the requested DB2 subsystem is active, control returns when CICS and DB2 have been connected. If the requested DB2 subsystem is not active, the response returned is 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.

## CEMT SET DB2CONN

### FORCENOTCON

FORCENOTCON force stops the connection by force purging transactions currently using DB2. Control is not returned until the connection is stopped.

### DB2GROUPIXD

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. DB2GROUPIXD can only be changed when CICS is not connected to a DB2 system. Specifying a DB2GROUPIXD 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 DSNX STRT command, this overrides any DB2GROUPIXD that is set in the installed DB2CONN definition. The DB2GROUPIXD 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 DB2GROUPIXD 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 DB2GROUPIXD in the installed DB2CONN definition to be blanked out, and the DB2GROUPIXD needs to be set again to use group attach. If a DB2GROUPIXD 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 DB2GROUPIXD 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.

### Nontermre1

specifies whether or not non-terminal transactions release threads for reuse at intermediate syncpoints. CVDA values are:

#### RELEASE

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

#### NORELEASE

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

### PLAN

specifies the name of the plan to be used for all threads in the pool. If PLAN is specified, PLANEXITNAME 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). If CICS is connected to DB2 Version 6 or later, the thread TCBs are CICS open L8 TCBs. If CICS is connected to DB2 Version 5 or earlier, the thread TCBs are private TCBs created by the CICS-DB2 Attachment Facility. CVDA values are:

**HIGH** Thread TCBs have a higher priority than the CICS QR TCB.

#### **EQUAL**

Thread TCBs have equal priority with the CICS QR TCB.

**LOW** Thread TCBs have a lower priority than the CICS QR TCB.

#### **PURGECYCLEM**

specifies in minutes the length of time of the protected thread purge cycle. The default is 0,30; that is, 30 seconds.

A protected thread is not terminated immediately when it is released. It is terminated only after two completed purge cycles, if it has not been reused in the meantime. Hence if the purge cycle is set to 30 seconds after it is released, a protected thread is purged 30 - 60 seconds after it is released. An unprotected thread is terminated when it is released (at syncpoint or end of task) if there are no other transactions waiting for a thread on that DB2ENTRY.

#### **PURGECYCLES**

specifies in seconds the length of time of the protected thread purge cycle. The default is 0,30; that is, 30 seconds.

A protected thread is not terminated immediately when it is released. It is terminated only after two completed purge cycles, if it has not been reused in the meantime. Hence if the purge cycle is set to 30 seconds after it is released, a protected thread is purged 30 - 60 seconds after it is released. An unprotected thread is terminated when it is released (at syncpoint or end of task) if there are no other transactions waiting for a thread on that DB2ENTRY.

#### **Resyncmember**

This applies only if you are using group attach, and specifies the strategy that CICS adopts if outstanding units of work are being held for the last DB2 data sharing group member to which CICS was connected. (Units of work which are shunted indoubt are not included in this process, because CICS itself is unable to resolve those units of work at this time. Resynchronization for those UOWs will occur when CICS has resynchronized with its remote coordinator.) CVDA values are:

#### **RESYNC**

CICS connects to the same DB2 data sharing group member to resynchronize the outstanding units of work.

#### **NORESYNC**

CICS makes one attempt to connect to the same DB2 data sharing group member, and if that attempt fails, CICS connects to any member of the DB2 data sharing group and issues a warning about the outstanding units of work.

#### **SECURITY**

specifies the security. CVDA value is:

### 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(xxxxxxx) REFRESH for RACF 2.1 or later

### SIgnid

specifies the authorization ID to be used by the CICS DB2 attachment when signing on to DB2 for pool and DB2ENTRY threads specifying AUTHTYPE(SIGN), and command threads specifying COMAUTHTYPE(CSIGN).

### Standbymode

specifies the action to be taken by the CICS DB2 attachment if DB2 is not active when an attempt is made to start the connection from CICS to DB2. CVDA values are:

#### NOCONNECT

The CICS DB2 attachment should terminate.

#### CONNECT

The CICS DB2 attachment goes into 'standby mode' to wait for DB2.

#### RECONNECT

The CICS DB2 attachment goes into 'standby mode' and waits for DB2. Having connected to DB2, if DB2 subsequently fails the CICS DB2 attachment reverts again to standby mode and subsequently reconnects to DB2 when it comes up again.

### STATSqueue

specifies the transient data destination for CICS DB2 attachment statistics produced when the CICS DB2 attachment is shutdown.

### TCBLimit

specifies the maximum number of TCBs that can be used to process DB2 requests. When connected to DB2 Version 5 or earlier, the CICS DB2 attachment facility creates the TCBs in the form of subtasks up to the limit specified by TCBLIMIT. Each of these subtasks identifies to DB2 and creates a connection into DB2. When connected to DB2 Version 6 or later, CICS creates open TCBs (up to the limit specified in the system initialization parameter MAXOPENTCBS). The TCBLIMIT attribute of the DB2CONN definition governs 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:

#### 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.

Note that when you change the value of AUTHID, AUTHType, COMAUTHID, COMAUTHType or Slnid, 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.

Note that 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 only be changed 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/exit will be determined using the new rules.

### CEMT SET DB2ENTRY

#### Function

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.

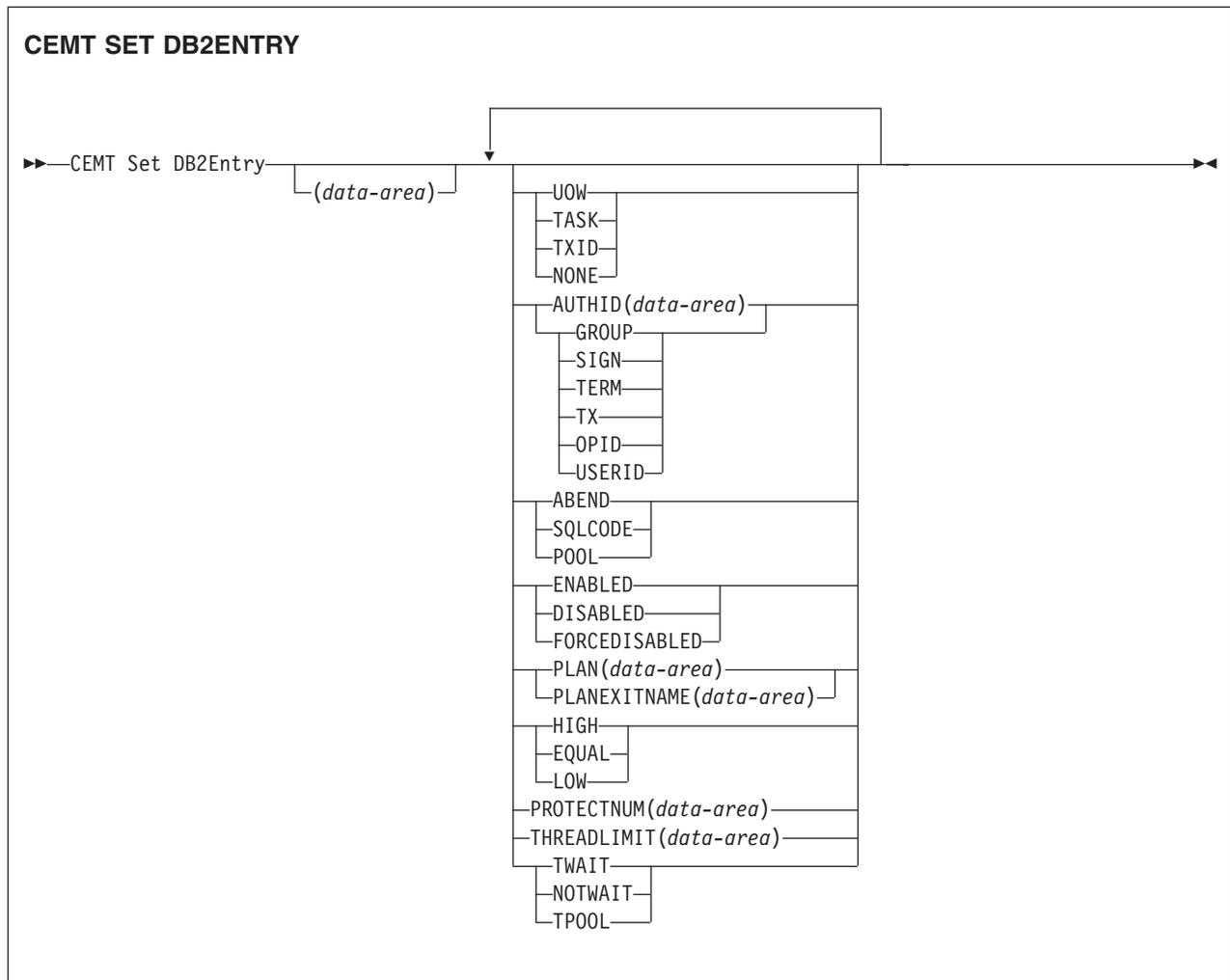
All parameters on SET DB2ENTRY can be set while the CICS DB2 attachment is active and the transactions are active.

#### 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.



## 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 at syncpoint, a

## CEMT SET DB2ENTRY

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 MSGQUEUE on the DB2CONN for each transaction routed to the pool.

**ABEND**

The CICS DB2 attachment facility abends the transaction. The abend code is AD26.

**SQLCODE**

An SQLCODE is returned to the application indicating that the DB2ENTRY is disabled.

**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). If CICS is connected to DB2 Version 6 or later, the thread TCBs are CICS open L8 TCBs. If CICS is connected to DB2 Version 5 or earlier, the thread TCBs are private TCBs created by the CICS-DB2 Attachment Facility. CVDA values are:

**HIGH** Thread TCBs have a higher priority than the CICS QR TCB.

**EQUAL**

Thread TCBs have equal priority with the CICS QR TCB.

## CEMT SET DB2ENTRY

**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

### Function

specify this 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.

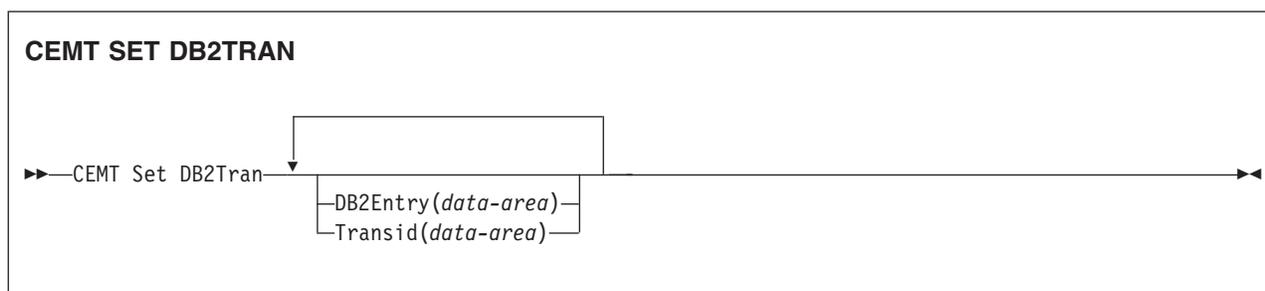
If a transid is specified on a DB2ENTRY when the DB2ENTRY is installed, CICS installs a DB2TRAN named DFHxxxx, where xxxx is the transid.

### 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 overwrite 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.



### 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 *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 DELETSHIPED

## CEMT SET DELETSHIPED

### Function

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 DELETSHIPED 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 DELETSHIPED (the minimum abbreviation is CEMT S DE). You get a display that lists the current status, similar to that obtained by CEMT INQUIRE DELETSHIPED. You can then tab to the highlighted or blank fields and overwrite them with the required values.
- Type CEMT SET DELETSHIPED (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 DELETSHIPED

▶▶—CEMT Set DEletshipped—┐Interval (value)┘┐IDle(value)┘▶▶

### 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

## CEMT SET DISPATCHER

### Function

Change the system values used by CICS dispatcher.

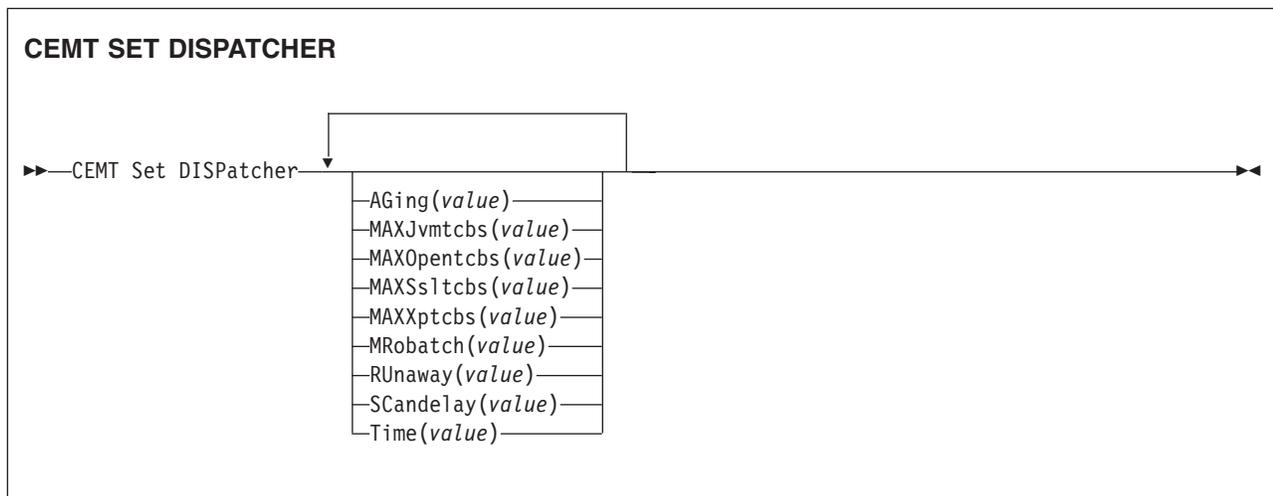
# **Note:** Except for MAXxxxTCBS you can also set these dispatcher system values using the CEMT SET SYSTEM command.

### 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 overtyping 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.



### 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.

**MAXS8tcbs** (*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.

## CEMT SET DISPATCHER

### 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 DSAS

### Function

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	
▶▶—CEMT Set DSAs	└─Dsalimit(value)┐ └─EDsalimit(value)┐

### 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

### Function

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

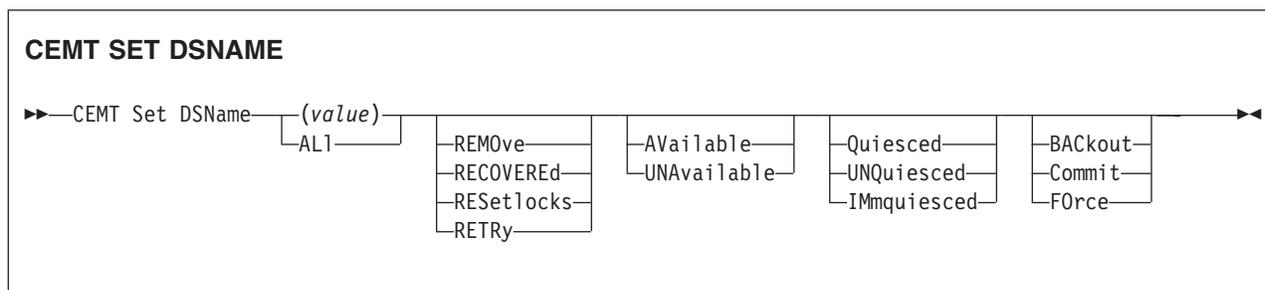
Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- 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(app1b.*) 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.



## 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

## CEMT SET DSNAME

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.

### Attention

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.

#### **Notes:**

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

## **CEMT SET DSNAME**

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

### Function

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	
▶▶—CEMT Set Dumps	▶▶
Open	Autoswitch
Closed	Noautoswitch
Switch	

### 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

## CEMT SET ENQMODEL

### Function

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

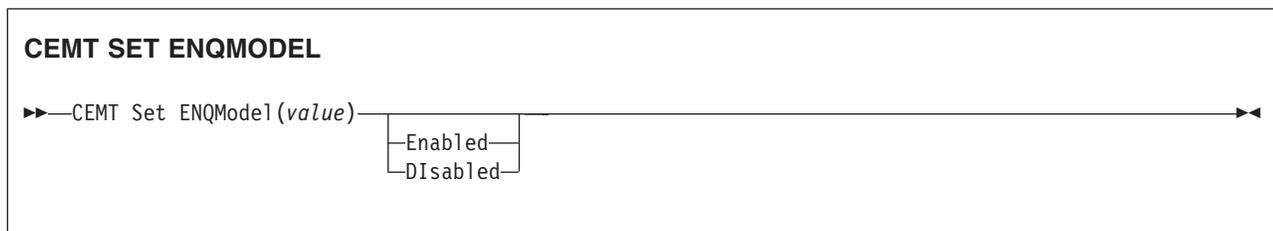
Matching ENQ requests will be rejected, and the issuing task abended abcode ENQ\_DISABLED. There are INSTALL/CREATE/DISCARD requests waiting to be processed.

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



### 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

## CEMT SET FILE

### Function

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 overtyping 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, `cent s fi a1 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					
CEMT Set File	(value) ALL	OPen CLosed ForcEclose	ENabled DISabled	REAd NORead	UPdate NOUpdate
ADdable NOAddable	BRowse NOBrowse	DElete NODElete	EXclusive NOEXclusive	EMptyreq NOEMptyreq	Old Share
RLs NOTR1s	DSname (value)	NOTTable CICstable USertable CFTable	LOAD NOLOAD	CFDtpool (value)	
Tablename (value)	CONtention LOCKing	Maxnumrecs (value)	Keylength (value)		
RECORDsize (value)					

### 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.

**DIabled**

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

## CEMT SET FILE

to be incremented by 1. You can use CEMT SET FILE DSNNAME with a null DSNNAME value to dissociate a file from its current data set.

**Note:** DSNNAME 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.

#### **Data integrity**

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.

**NOTRIs (vsam only)**

The file is defined to be accessed in a non-RLS mode (LSR or NSR).

## CEMT SET FILE

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.

### **01d**

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

### Function

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, `cent 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			
▶▶—CEMT Set Gtftrace	▶▶		
<table border="1"> <tr> <td>STArt</td> </tr> <tr> <td>STOp</td> </tr> </table>	STArt	STOp	
STArt			
STOp			

### 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

### CEMT SET HOST

#### Function

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

► CEMT Set Host (value) Enabled Disabled ◄

#### Options

##### (value)

specifies the name of the virtual host to which this command applies. 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.research.ibm.com, CICS would create a virtual host with the name www.research.ibm.com. 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

### Function

Switch CICS internal tracing on or off.

### Description

SET INTTRACE allows you to switch CICS internal tracing on and off.

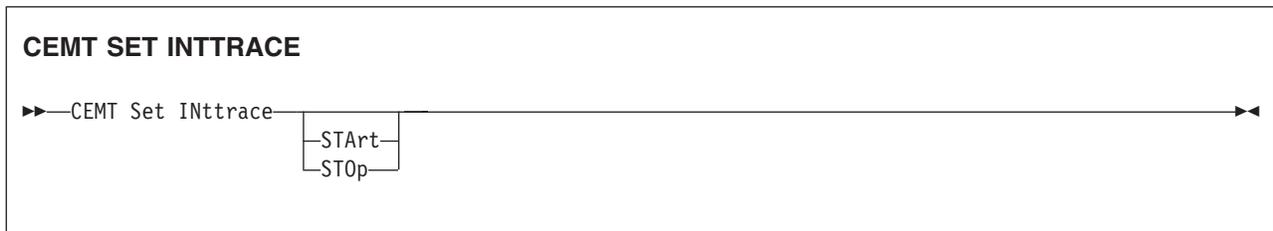
This command is intended for use at terminals that do not support the full-screen tracing transaction CETR.

### 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 overwrite it with the required value.
- Type CEMT SET INTTRACE (CEMT S IN) followed by the attribute setting that you wish to change. For example, `cent 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.



### Options

#### STArt

CICS internal tracing is started.

#### STOp

CICS internal tracing is stopped.

## CEMT SET IRC

## CEMT SET IRC

### Function

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:

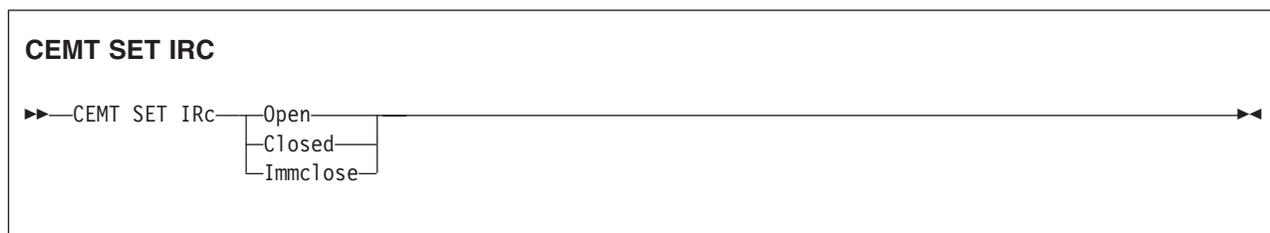
```
CEMT SET IRC CLOSED
CEDA INSTALL GROUP(groupname)
CEMT SET IRC OPEN
```

### 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 overwrite 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.



### 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 `INSERVICEIOUTSERVICE` settings on the connection names.

The DFHIR3780 message and CICS interregion communication program (IRP) return and reason codes are displayed if IRC fails to open. For a description of DFHIR3780, see the *CICS Messages and Codes* manual.

## CEMT SET JOURNALNAME

### Function

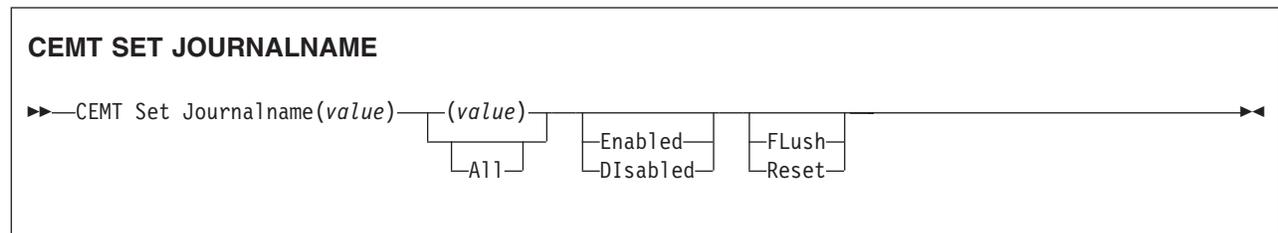
Enable or disable a CICS user journal.

### 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



### 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.

#### A11

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

### Function

Change the status of a JVM pool.

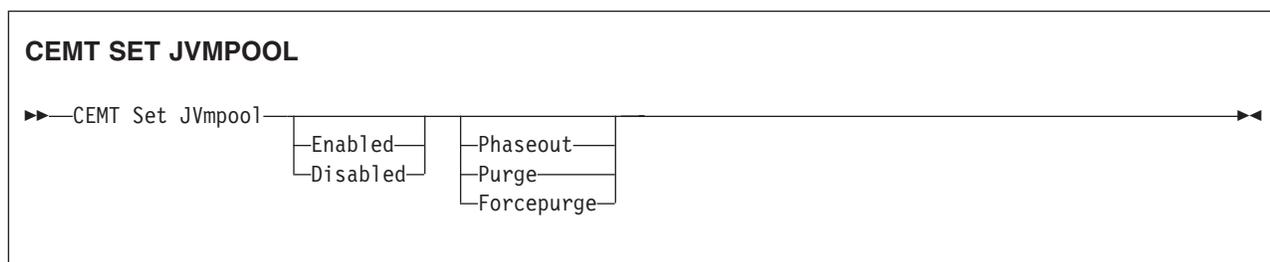
### Description

The SET JVMPOOL command allows you to enable or disable the JVM pool, or to terminate the pool altogether. When you terminate the pool, all the JVMs in the pool, both worker JVMs using the shared class cache and JVMs running independently of the shared class cache, are terminated. No more JVMs can use the existing shared class cache, and it is deleted when all the worker JVMs that were dependent on it have been terminated. If autostart is enabled for the shared class cache, a new shared class cache is started as soon as a new JVM requests it.

### 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 overwrite 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.



### 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 is 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 in the pool are terminated by the SET TASK FORCEPURGE mechanism. The shared class cache is deleted when all the worker JVMs that were dependent on it have been terminated.

#### Phaseout

All JVMs in the pool are marked for deletion. The JVMs are actually deleted when they finish running their current Java program. The shared class cache is deleted when all the worker JVMs that were dependent on it have been terminated.

#### Purge

All tasks using JVMs in the pool are terminated by the SET TASK PURGE

mechanism, and the JVMs are terminated. The shared class cache is deleted when all the worker JVMs that were dependent on it have been terminated.

## CEMT SET LINE

## CEMT SET LINE

### Function

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.

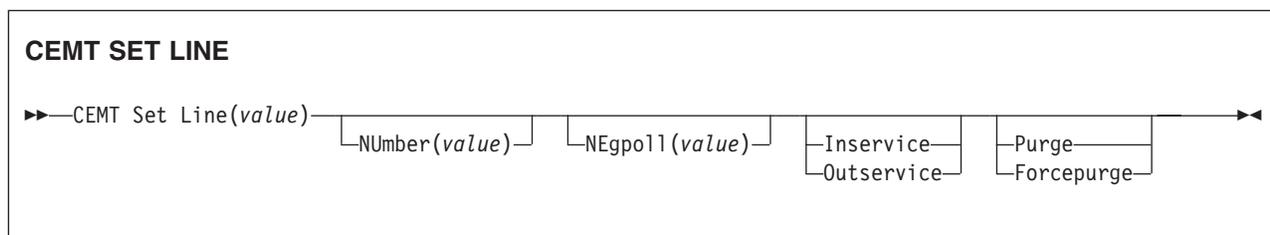
You can specify a list of identifiers (see “List of resource identifiers” on page 140), but you cannot use the symbols \* and + to specify a family of identifiers.

### 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 overwrite 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.



### 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

## CEMT SET MODENAME

### Function

Change the status of a mode name.

### Description

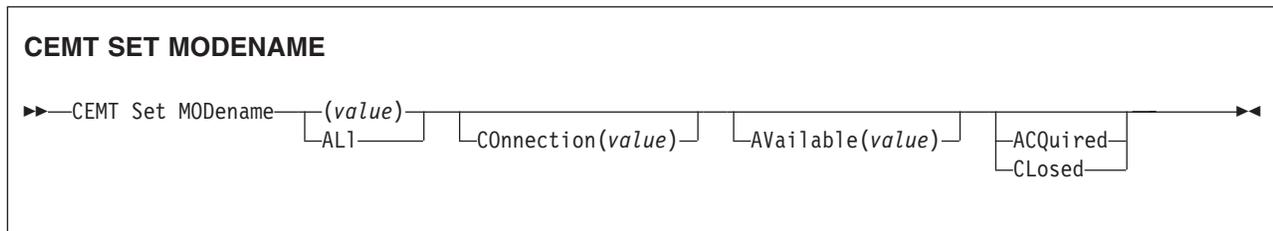
The SET MODENAME command is used to reset the connection to a remote system or to another CICS region, change the number of sessions available, and acquire or close the sessions associated with the mode name.

### 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 overwrite 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.



### 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.

**COnection (value)**

specifies an identifier (1–4 characters) defined for an IRC or ISC connection.

# CEMT SET MONITOR

### Function

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*.

### Notes:

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.

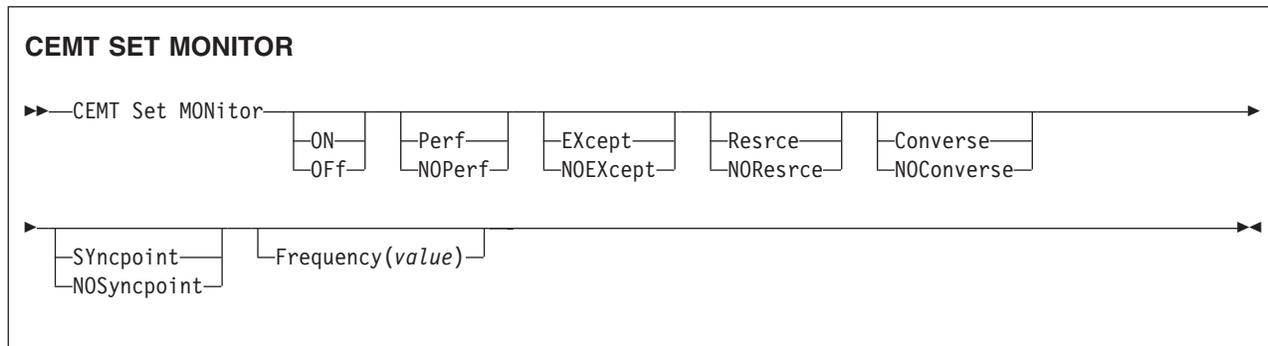
- c. Exception class, transaction resource class, and performance class data are gathered during the life of a running transaction. When you change these settings, you should be aware of their impact on transactions that are running. For example, if a transaction is started when the performance class 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.

### 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 overwrite 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.



## Options

### 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).

### 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.

## CEMT SET MONITOR

### **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

### **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.

**P**Ageable (*vtam only*)

Pages after the first in a series are to be written to the terminal only if you request them to be.

**P**riority (*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.

**P**Urge

Tasks running on this terminal are abnormally terminated, but task termination occurs only if system integrity can be maintained.

**R**ELeased (*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.

**T**Ti (*vtam only*)

This terminal can be used by transactions.

## CEMT SET PIPELINE

### CEMT SET PIPELINE

#### Function

Enable or disable a PIPELINE.

#### Description

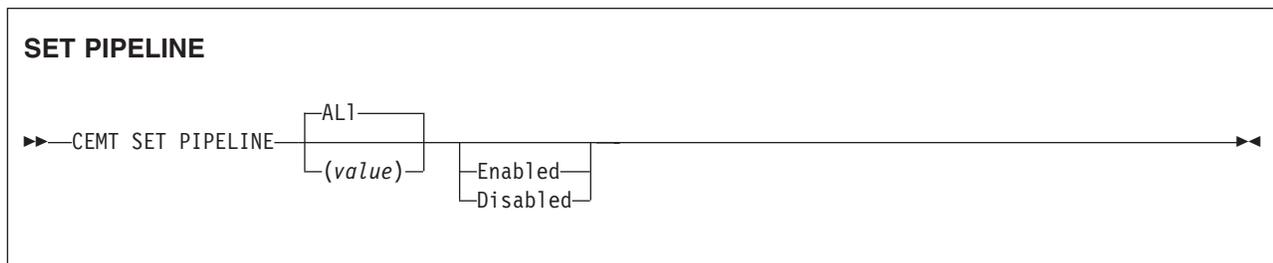
If the optional attributes are omitted, the command has no effect.

#### 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 all 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.



**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.

## CEMT SET PROCESSTYPE

### Function

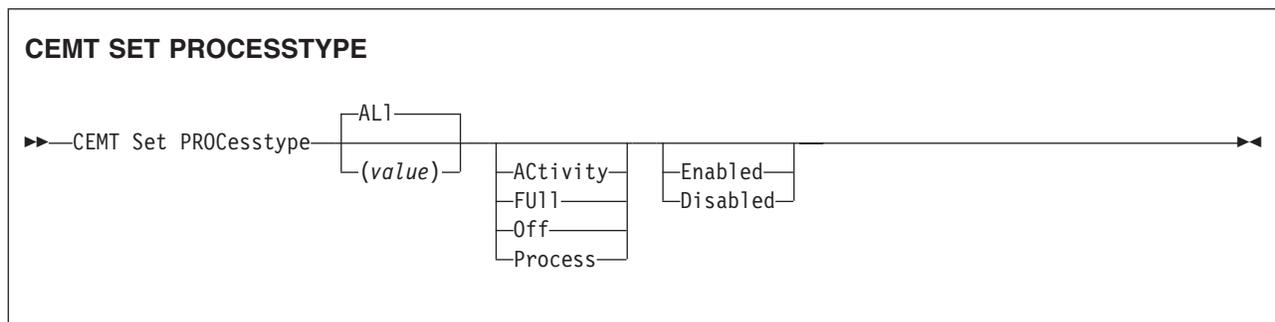
Change the attributes of a CICS business transaction services process-type.

### 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



### 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.

## CEMT SET PROCESSTYPE

For details of the records that are written from the process, activity primary, and activity secondary audit points, see the *CICS Business Transaction Services* manual.

### AL1

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

### Function

Change the attributes of installed programs.

### Description

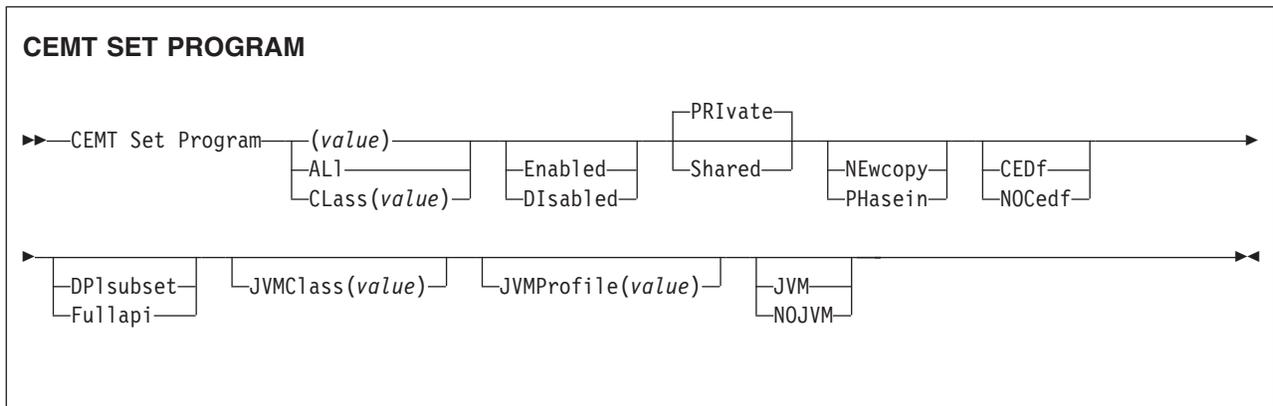
Only programs that have been defined in the CICS system definition (CSD) file and installed on the running CICS system are accessible through CEMT.

### 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 pgmid, 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 overwrite 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(pgmid) 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.



### 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.

## CEMT SET PROGRAM

### **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 "clasic".

If the clasic specified is the suffix of a PLT that does not have a program definition, the INQUIRE PROGRAM CLASS(clasic) 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.

### **DPIsubset**

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 under the control of a Java Virtual Machine (JVM).

### **JVMClass** (*value*)

specifies the name of the main JVM 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 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 HFS 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 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 JVM program.

### **NOJvm**

The program is not to run under the control of a JVM. Any value in the JVMCLASS option is ignored and the runtime environment of the program is unknown until it is next loaded by CICS, at which point its language, and hence whether it will run with LE370 support, will be 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 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 JVM program.

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 library. 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 JVM program.

## CEMT SET QUEUE

## CEMT SET QUEUE

### Comments

See “CEMT SET TDQUEUE” on page 500. (QUEUE is accepted for INQUIRE commands, but the correct keyword is TDQUEUE.)

## CEMT SET STATISTICS

### Function

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**

▶▶—CEMT Set Statistics—▶▶

	ON	RESetnow	RECORDnow	Interval (hhmmss)
	OFF			

▶—Endofday (hhmmss)—▶

### 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).

## CEMT SET STATISTICS

### 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.

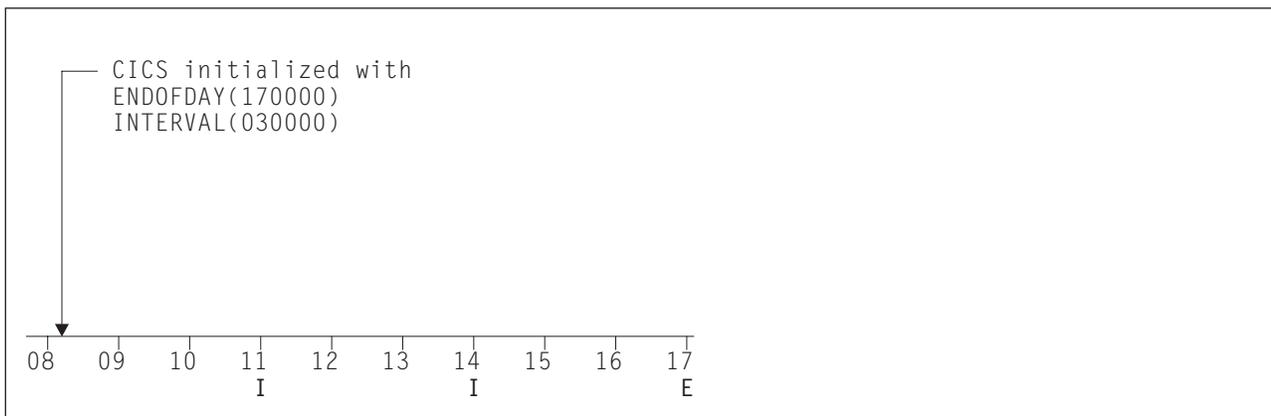


Figure 186. Changing the INTERVAL or ENDOFDAY value

**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

### Function

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 the *CICS Messages and Codes* manual 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
```

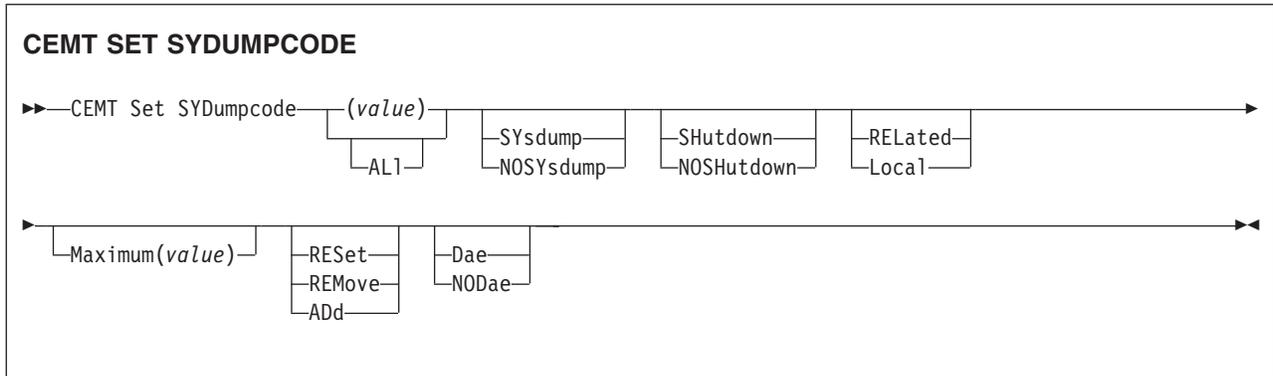
To change the maximum count to 10:

```
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.



## Options

### (*value*)

specifies an 8-character system dump code. See the *CICS Messages and Codes* manual 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* manual.

### 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.

## CEMT SET SYDUMPCODE

### **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

### Function

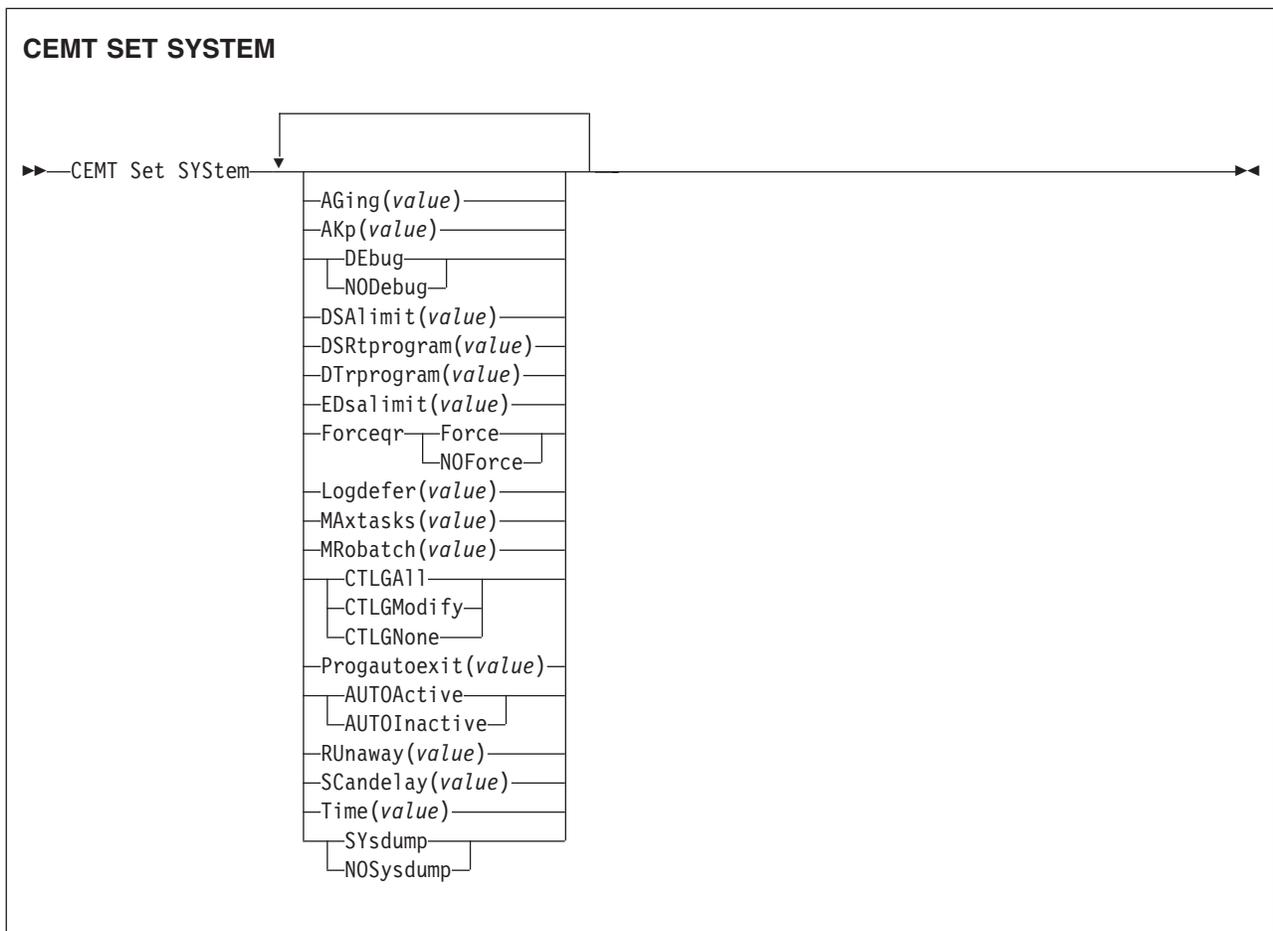
Change the system attributes.

### 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 overtyping 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.



### Options

#### AGing (*value*)

specifies the time factor for CICS to use in the priority aging algorithm for

## CEMT SET SYSTEM

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.

### **CTLGA11**

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:

## CEMT SET SYSTEM

- 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 using 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

### **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 TCPIP

## CEMT SET TCPIP

### Function

Open or Close CICS TCP/IP Service.

### Description

SET TCPIP allows you to open or close CICS internal sockets support.

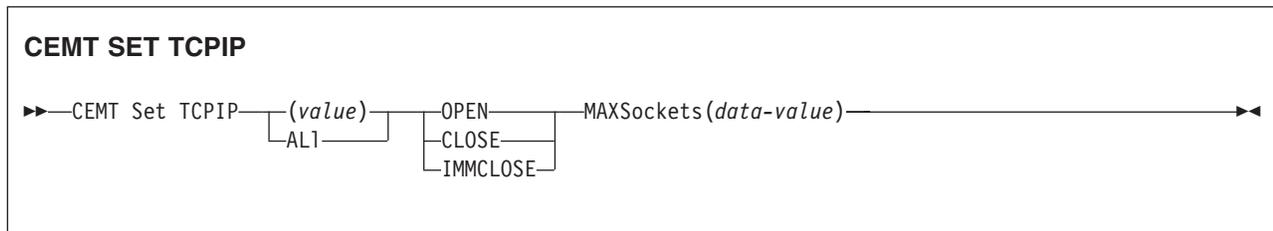
**Note:** This command has no effect on the sockets support provided by the TCP/IP for OS/390 CICS Sockets Feature.

### 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 overwrite 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.



### 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 TCPIP SERVICE

### Function

Modify the status of a service using CICS internal TCP/IP support

### Description

SET TCPIP SERVICE allows you to update the information relating to a service which uses CICS internal TCP/IP support.

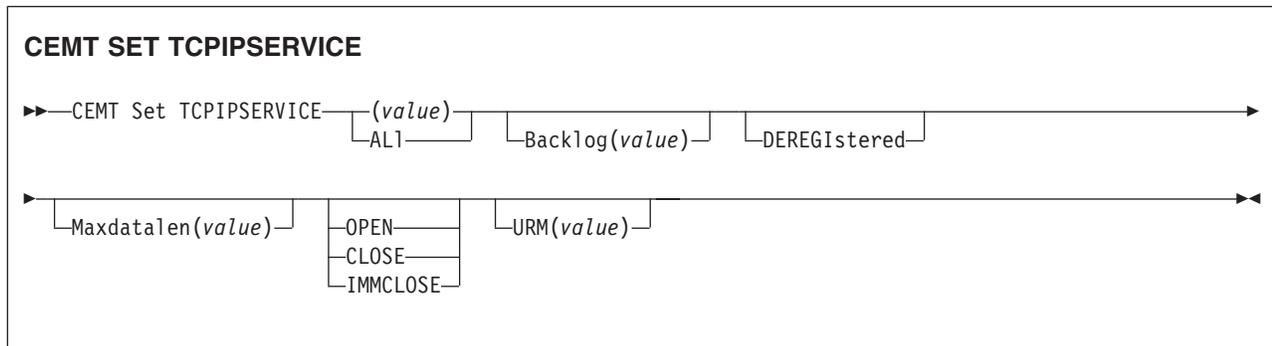
**Note:** This command has no effect on the sockets support provided by the TCP/IP for OS/390 CICS Sockets Feature.

### Syntax

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

- Type CEMT SET TCPIP SERVICE (the minimum abbreviation is CEMT S TCPIPS). You get a display that lists the current status, similar to that obtained by CEMT INQUIRE TCPIP SERVICE. 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.



### 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 TCPIP SERVICE. This can be done independently of changing the open or closed status of the service.

To account for timing delays in the deregister request reaching the WLM and the DNS updating its tables, it is advisable to deregister a service before setting it closed. This ensures that client applications do not encounter "Connection Refused" situations during the time between the deregister call being issued and the DNS server actually updating its tables. The valid CVDA values are:

**DEREGISTERED**

causes CICS to deregister the group name specified by the DNSGROUP attribute of this TCPIPSERVICE. The OS/390 WLM macro IWMSRDRS is called and CICS will no longer be a part of the DNS connection optimization. Any other TCPIP SERVICES 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.

**Note:** This program and any programs it invokes must be installed before they can be used in the program autoinstall process. You can do this either with explicit PROGRAM definitions or by autoinstall when some other autoinstall program is in force. Otherwise the program autoinstall process fails when it is next used, and CICS makes it inactive.

# CEMT SET TDQUEUE

### Function

Change 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
- 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.

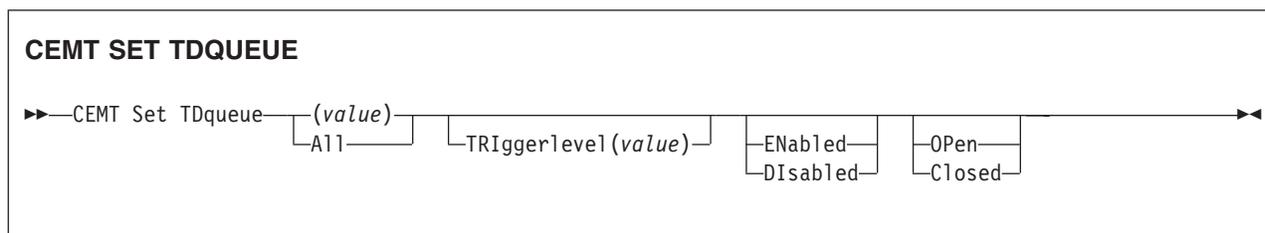
**Note:** If a task updates a logically recoverable transient data queue, and attempts to disable the queue and alter an attribute of the queue (for example, ATITRANID) within the same UOW, the call fails. This is because the UOW is a user of the queue, and the queue enters a “disable pending” state. The SET operation on the queue attribute, in this case ATITRANID, fails. The queue does not become fully disabled until the UOW commits or backs out at syncpoint. You are recommended to issue an EXEC CICS SYNCPOINT command before attempting to update the queue attribute (ATITRANID) using SET TDQUEUE.

## Syntax

Press the Clear key to clear the screen. There are two ways of commencing 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 overwrite 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.



## Options

### (*value*)

specifies the identifier (1-4 characters) 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

#### Function

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 473.

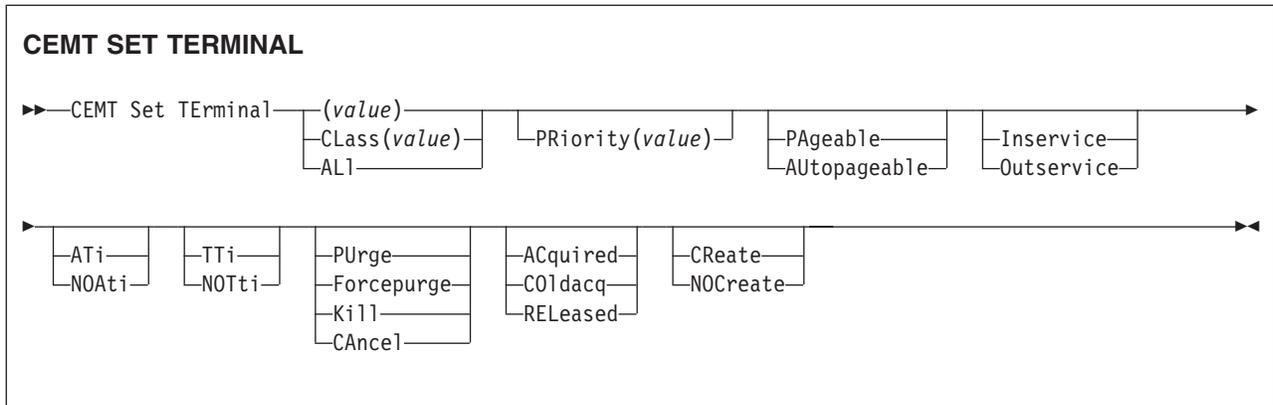
The message “TEMP OUT SERVICE” can be displayed if recovery is in progress for the terminal. The “TEMP OUT SERVICE” indicator is reset by simply refreshing the screen, or by overtyping the “INS” field with “INS” or “OUT”. If “TEMP OUT SERVICE” continues to be displayed, the cause of the indicator should 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.



## 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.

### **AL1**

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.

### **CAnce1**

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).

### **CO1dacq** (*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

## CEMT SET TERMINAL

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.

### **CR**eat (vtam only)

If the terminal is not in session, CICS is to acquire it if it is needed to satisfy an ATI request.

### **Force**purge

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.

### **In**service

The terminal is available for use. For VTAM, INSERVICE means that the terminal can be ACQUIRED.

### **K**ill

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.

### **NO**Ati (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.

### **NO**Create (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.

### **NO**Tti (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.

### **Out**service

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.

**P***Ageable* (*vtam only*)

Pages after the first in a series are to be written to the terminal only if you request them to be.

**P***riority* (*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.

**P***Urge*

Any transaction running with this terminal is purged only if system and data integrity can be maintained.

**R***Eleased* (*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.

**T***Ti* (*vtam only*)

This terminal can be used by transactions.

## CEMT SET TRANSACTION

## CEMT SET TRANSACTION

### Function

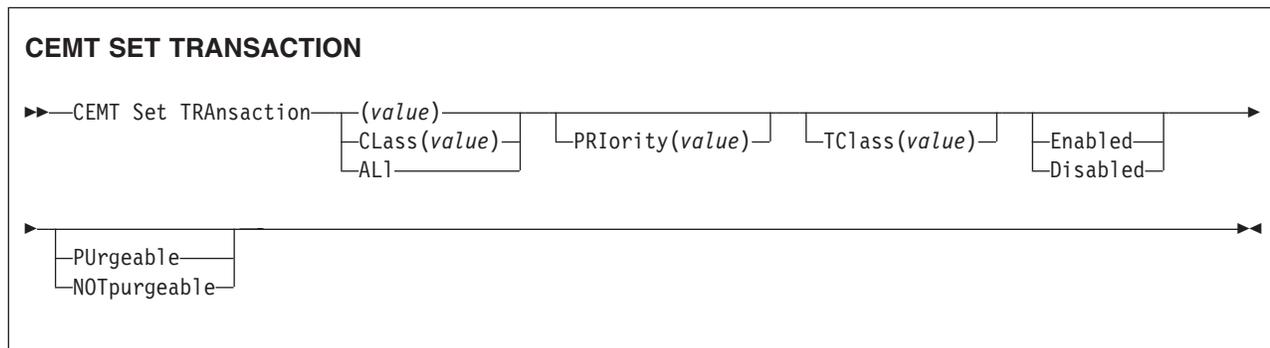
Change some of the attributes of a selected transaction.

### 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 overwrite 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, cent 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.



### 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 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

**Function**

Change the status of a transaction dump.

**Description**

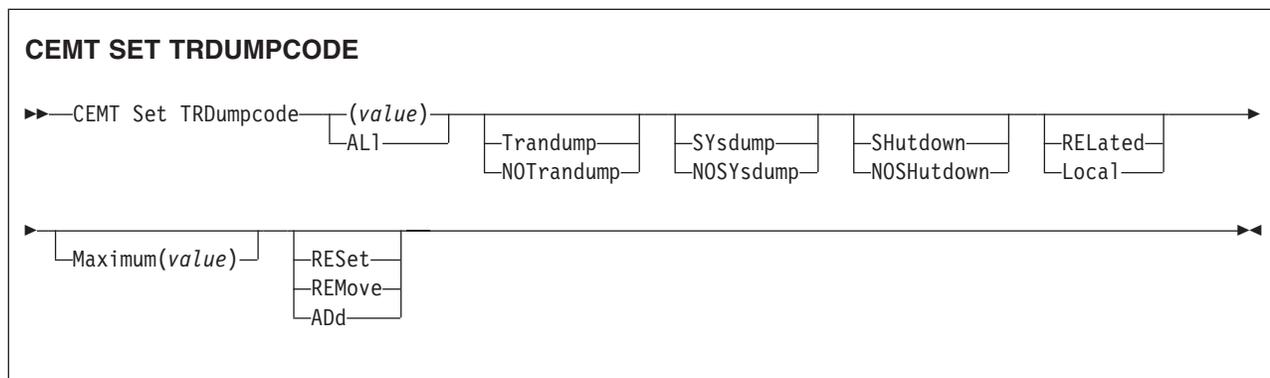
Using the SET function, you can change entries in the transaction dump table to request a transaction dump. You can also set the maximum number of dumps that can be taken. See the *CICS Problem Determination Guide* for information on using dumps.

**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 the *CICS Messages and Codes* manual for a list of transaction dump codes.

**ADd**

The specified transaction dump code is to be added to the dump code table.

**AL1**

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

## CEMT SET TSQUEUE

### Function

Delete a temporary storage queue (TS queue).

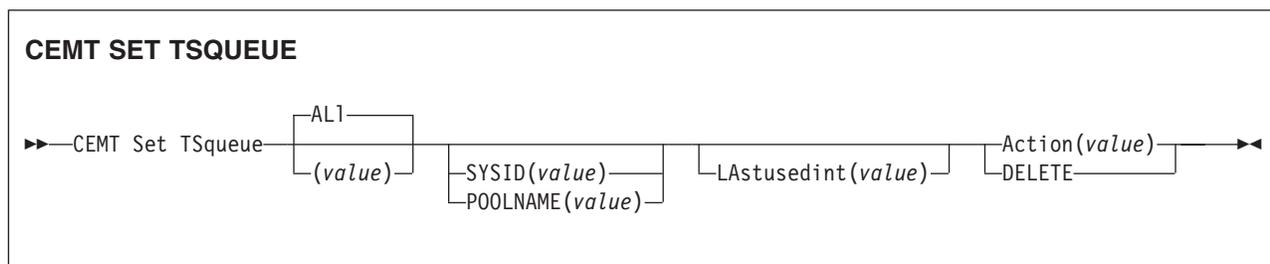
This section applies also to the alternative command, **CEMT SET TSQNAME**. Use either to delete temporary storage queues (TS queues) with names up to 8 characters long. Use SET TSQNAME to delete temporary storage queues (TS queues) with names up to 16 characters long.

### 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



### Options

*(value)*

specifies the identifier of the TS queue

**AL1**

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

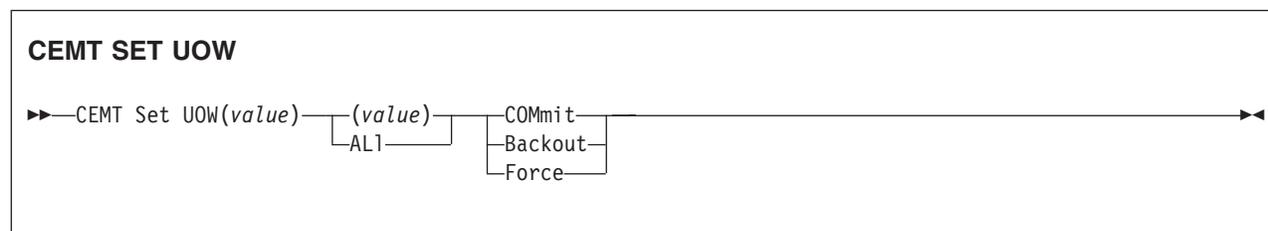
### Function

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



### Options

#### (*value*)

specifies the 16-character identifier of the UOW to be committed, backed out, or forced.

#### AL1

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

## CEMT SET UOWLINK

### Function

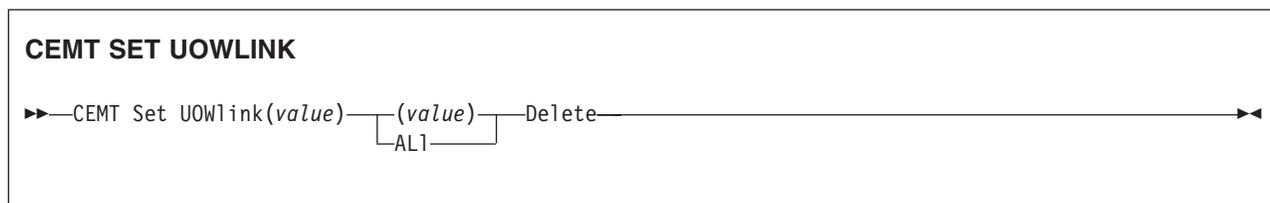
Delete a link to a unit of work that was created by a connection that has since been discarded.

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



### Options

#### (*value*)

specifies the 4-character identifier of the UOW-connection dependency (the UOW-link) to which this command applies.

#### AL1

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

### Function

Enable or disable a URIMAP definition, and apply or remove redirection for a URIMAP definition.

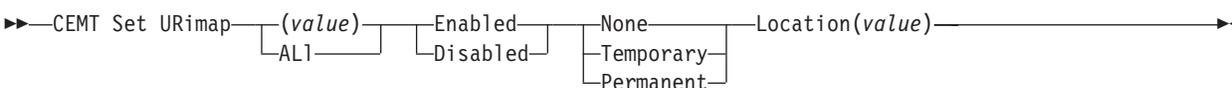
### Description

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.

## CEMT SET URIMAP

|                   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

### Function

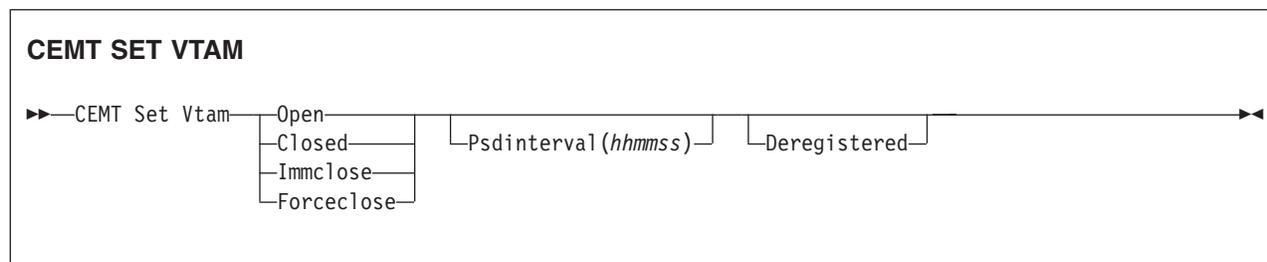
Open or close the VTAM ACB, set the persistent session delay interval (PSDI), or deregister CICS from membership of a generic resource.

### 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 overwrite 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.



### 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.

## CEMT SET VTAM

### **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*

<b>Message</b>	<b>RESP/RESP2</b>
INVALID PSDINT	RESP=INVREQ, RESP2=4
NOT WITH XRF	RESP=INVREQ, RESP2=8
SETLOGON FAILURE	RESP=INVREQ, RESP2=9
BACK LEVEL VTAM	RESP=INVREQ, RESP2=10
ACB CLOSED	RESP=INVREQ, RESP2=12
RECOVERY ERROR	All other cases of INVREQ



## CEMT SET WEBSERVICE

### Function

Set the validation status of a WEBSERVICE.

### Description

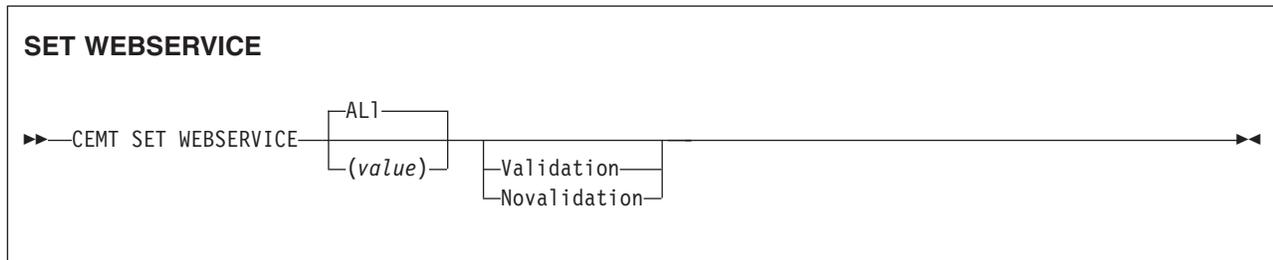
If the optional attributes are omitted, the command has no effect.

### 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 a1 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.



**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

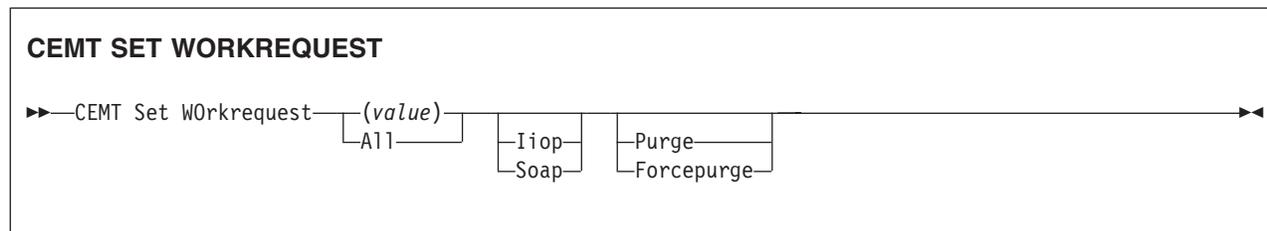
SET WORKREQUEST enables you to terminate specific work requests. You cannot specify a list of identifiers, nor can you use the symbols \* and + to specify a family of work requests.

### 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 overwrite 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.



### 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.

## CEMT SET WORKREQUEST

- |                   **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 15. 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 522) 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 >

RESPONSE: NORMAL                                SYSID=HT61 APPLID=CICSHT61
PF 1 HELP          3 END          5 VAR          TIME: 13.47.07 DATE: 02.01.01
                                                         9 MSG
```

Figure 187. 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.

**Acq** CICS is in session with the logical unit represented by this terminal. You cannot overwrite 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 overwrite this value.

**Ins** Your terminal is INSERVICE. You cannot overwrite 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.

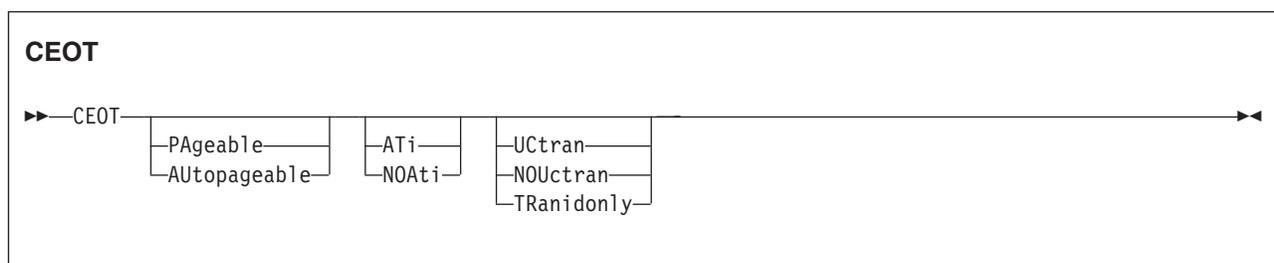
**Tra** (*ceot*)  
displays the transaction that is running on your terminal. This is always CEOT.

**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.

**Notes:**

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 16. 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:



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.

**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 17. 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 188).

```
DFHZC2312 *** WELCOME TO CICS *** 12:56:28

          *****\ *****\ *****\ *****\ (R)
*****\ *****\ *****\ *****\
**\|\|\|\|\**\ **\|\|\|\|\**\ **\|\|\|\|\**\ **\|\|\|\|\**\
**\  \  \  \  \  \  \  \  \  \  \  \  \  \  \  \  \  \  \  \  \
**\      **\      **\      *****\
**\      **\      **\      *****\
**\      **\      **\      \|\|\|\|\**\
**\  **\  **\  **\  **\  **\  **\  **\
*****\ *****\ *****\ *****\
*****\|\ *****\|\ *****\|\ *****\|\
\|\|\|\|\  \|\|\|\|\  \|\|\|\|\  \|\|\|\|\
```

Figure 188. 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 189 on page 528.

The user ID specified for this transaction is used in error messages and so on, to identify the user.

### Notes:

1. If you sign on at a terminal that is already signed on, CSSN 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 530.
2. For numeric-only devices or for the IBM 2741 Communication Terminal, use the signon procedure described in “Non-3270 family terminal” on page 529.
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.
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 189. 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.

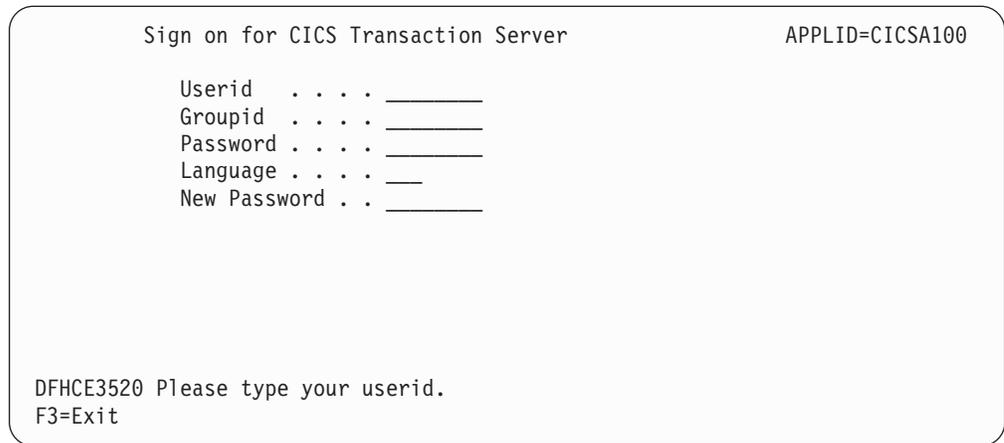


Figure 190. 40 x 12 screen

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**

▶—CESN USERID=uuuuuuuu—CESN—USERID=uuuuuuuu—GROUPID=gggggggg—,LANGUAGE=lll—▶

(1)

▶—,PS=pppppppp—,NEWPS=password—▶

**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=111**

enables you to choose a national language (from those that are available at your installation) for terminal directed messages. "111", 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 Transaction Server for z/OS, 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 18. CEST—supervisory terminal

Use CEST to invoke a subset of the master terminal (CEMT) transaction. The CEST INQISET 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 INQUIREISET LINE, NETNAME, TASK, and TERMINAL.



## Chapter 19. 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     ==>>>  STARTED   STArted, STOpped
Internal Trace Table Size ==>>>  16      K      16K - 1048576K

Auxiliary Trace Status    ==>>>  STOPPED   STArted, STOpped, Paused
Auxiliary Trace Dataset   ==>>>  B        A, B
Auxiliary Switch Status   ==>>>  NO       NO, NExt, All

GTF Trace Status         ==>>>  STOPPED   STArted, STOpped

Master System Trace Flag  ==>>>  ON       ON, Off
Master User Trace Flag    ==>>>  ON       ON, Off

When finished, press ENTER.

PF1=Help   3=Quit   4=Components   5=Ter/Trn   6=JVM   9=Error List
  
```

Figure 191. CETR transaction: initial screen

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 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. The system  
 # command DISPLAY ASM MVS can be used 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 192. 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:

<b>Component</b>	<b>Title</b>
AP	Application domain
BA	CICS business application manager for BTS
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	IIOF 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

Component	Title
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
US	User domain
WB	Web interface
XM	Transaction manager
XS	Security manager domain

**Notes:**

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 “Using JVM trace options” on page 538.
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 193. 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 192 on page 534, 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 193, 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.

---

## Using JVM trace options

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.)

You can use these screens 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 194. 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 195. CETR transaction: JVM trace options screen 2 of 2

The default JVM trace options that are provided in CICS use the JVM trace point level specifications. The default settings for JVM Level 0 trace, JVM Level 1 trace, and JVM Level 2 trace specify LEVEL0, LEVEL1, and LEVEL2 respectively, so they map to the Level 0, Level 1 and Level 2 trace point levels for JVMs. A Level 0 trace point is very important, and this classification is reserved for extraordinary events and errors. Note that unlike CICS exception trace, which cannot be switched off, the JVM Level 0 trace is normally switched off unless JVM tracing is required. The

Level 1 trace points and Level 2 trace points provide deeper levels of tracing. The JVM trace point levels go up to Level 9, which provide in-depth component detail.

It is suggested that you keep the CICS-supplied level specifications, but if you find that another JVM trace point level is more useful for your purposes than one of the default levels, you could change the level specification to map to your preferred JVM trace point level (for example, you could specify LEVEL5 instead of LEVEL2 for the JVMLEVEL2TRACE option). Note that enabling a JVM trace point level enables that level and all levels above it, so for example, if you activate JVM Level 1 trace for a particular transaction, you receive Level 0 trace points for that transaction as well. This means that you only need to activate the deepest level of tracing that you require, and the other levels are activated as well.

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 which use multiple trace point levels, or if you do not want to use trace point levels at all in your specification, use the JVMUSERTRACE option to create a trace option string that includes the parameters of your choice. “Defining tracing for JVMs” in 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. 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, Version 1.4.2 Diagnostics Guide, SC34-6358*, which is available to download from [www.ibm.com/developerworks/java/jdk/diagnosis/](http://www.ibm.com/developerworks/java/jdk/diagnosis/).

Trace levels 29–32 for the SJ component correspond to JVM Level 0, Level 1, Level 2, and User trace respectively. When you activate JVM trace, each JVM trace point that is generated appears as an instance of CICS trace point SJ 4D01. If the JVM trace facility fails, CICS issues the trace point SJ 4D00.

As well as using CETR, you can also control JVM trace by using:

- The CICS system initialization parameters JVMLEVEL0TRACE, JVMLEVEL1TRACE, JVMLEVEL2TRACE, and JVMUSERTRACE, which you can use to set the default JVM trace options for the CICS region, and the SPCTRSJ and STNTRSJ system initialization parameters, which you can use to activate JVM tracing. You can only supply these parameters at CICS startup time; you cannot define them in the DFHSIT macro.
- The EXEC CICS INQUIRE JVMPOOL and EXEC CICS SET JVMPOOL commands (but not their CEMT equivalents), which you can use to change the current JVM trace options for the CICS region.
- The EXEC CICS SET TRACETYPE command, which you can use to activate JVM tracing for the CICS region.
- The **ibm.dg.trc.external** system property in the JVM properties file that is referenced by the JVM profile for a JVM, which you can use to set initial trace options for a particular JVM. This enables you to trace a JVM during its whole lifetime, including start-up and reset as well as the periods when it is being used by a transaction. It has to be used with care, as JVM tracing can produce large amounts of output in a very short time. “Defining tracing for JVMs” in the *CICS Problem Determination Guide* has information about the JVM trace options, which are specified in the same format for this system property as they are for the methods available in the CICS interface.

When CICS starts to use or re-use a JVM, it ensures that any trace options that you have set and activated using CETR are applied. Activating or deactivating a

trace option using CETR overrides any setting for that trace option in the **ibm.dg.trc.external** system property. For example, a trace option that is activated in the system property, but deactivated using CETR, will be deactivated when CICS starts to use or re-use the JVM. If you use CETR to activate any trace options that are not referred to in the **ibm.dg.trc.external** system property, the trace options that you have specified in CETR are *added to* any trace options that you have set using the **ibm.dg.trc.external** system property. The trace output will then reflect all the trace options that you requested in both CETR and the system property.

"Controlling tracing for JVMs" in *Java Applications in CICS* tells you about these alternative ways to control tracing for your JVMs.

---

## 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 196. 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 ST0 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 197. 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-extends.
- 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 198. 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 199. 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 200. 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 201. 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 202. 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 203. 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 204. 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 205. 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 206. 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 207. 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 208. 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 209. 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 210. 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 211. 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 212. CETR transaction: JVM trace options HELP panel (3)



---

## Chapter 20. 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.



---

## Chapter 21. 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.

---

## Running the CIND in-doubt tool

1. Change the transaction class of the transactions you want to go in-doubt to DFHTCIND. (You can do this using CEMT or CEDA.)
2. Type CIND ON. This activates the tool and enables a task-related user exit for task start. You should get the following message:  

```
DFHIN1001 date time applid termid userid.  
The in-doubt tool is now active for  
DFHTCIND tranclass transactions.
```

For every transaction that is started, the task-related user exit inquires whether it is in transaction class DFHTCIND. If it is, the task-related user exit issues a call to the CICS recovery manager to add a link to CIND, acting as a CICS coordinator system called DFHINDSP.
3. Run your transactions. If the transactions update recoverable resources and the TRANSACTION resource definition specifies the INDOUBT attribute WAIT(YES), they fail in-doubt. The transaction abends with abend code ASP1, and the UOW is shunted. You can use the CEMT INQUIRE UOW SHUNTED command to display the shunted UOWs.  

A transaction that updates non-recoverable resources, or that issues only READ requests abends with abend code ASPR. However, the UOW is not shunted, because it is read-only, so no syncpoint resynchronization is required with remote coordinator DFHINDSP.
4. Type CIND OFF. This disables the task-related user exit and stops any further invocations of your transactions going in-doubt. You should get the following message:  

```
DFHIN1004 date time applid termid userid.  
The in-doubt tool is no longer active for  
DFHTCIND tranclass transactions.
```
5. Type CIND RESYNC COMMIT or CIND RESYNC BACKOUT. This simulates a coordinator CICS system reconnecting to your CICS system. In this case, the coordinator is DFHINDSP. You should get the following message:  

```
DFHIN1007 date time applid termid userid.  
Initiation of resynchronization for units of work  
awaiting coordinator DFHINDSP is now complete.
```

The CICS recovery manager domain unshunts all UOWs that were previously shunted because of the tool. The commit or backout specifies what happened in the dummy coordinator DFHINDSP, and hence what the CICS recovery manager domain instructs its subordinates to do when the unshunt takes place. For each UOW unshunted, message DFHIN1012 is sent to CSMT detailing the UOW status as defined by coordinator DFHINDSP, and the UOW status held by the CICS recovery manager. Both states should be the same.

You can use the CIND RESYNC COMMIT command or the CIND RESYNC BACKOUT command to simulate what happens when UOWs are not synchronized. For example, a transaction may be run with CIND activated and with a TRANSACTION resource definition that specifies the INDOUBT attributes WAIT(NO) and WAITACTION(COMMIT). In this case, the UOW is not shunted but is forced to commit. The CICS recovery manager domain keeps a record of the UOW in order to resynchronize with the coordinator. CEMT shows the UOW in committed status waiting for the return of the coordinator. If a CIND RESYNC BACKOUT command is issued at this point, the DFHIN1012 message output to CSMT shows the UOW to be out of sync, as does message DFHRM0111 output by CICS recovery manager domain.

## 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 in-doubt tool” on page 554. 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 22. CLER—Language Environment run-time options

The CLER transaction is provided by Language Environment and is not shipped as part of the CICS Transaction Server product. CLER is available with OS/390 2.8 Language Environment and later (see Language Environment APAR PQ38838).

CLER enables you to:

- Display the Language Environment run-time options
- Write the run-time options to the CESE queue for printing
- Modify a subset of the run-time options

CLER is a conversational transactional that presents the display panel shown in Figure 213.

```
CLER                                     CICS IYCLZCFE

                                LANGUAGE ENVIRONMENT REGION LEVEL RUNTIME OPTIONS

TYPE IN YOUR CHOICES.

RUNTIME OPTION      CHOICE      POSSIBLE CHOICES

TRAP                ==> OFF        ON, OFF
RPTOPTS             ==> OFF        ON, OFF
RPTSTG              ==> OFF        ON, OFF
ALL31               ==> OFF        ON, OFF
CBLPSHPOP           ==> OFF        ON, OFF
CHECK               ==> ON         ON, OFF
INFMSGFILTER        ==> OFF        ON, OFF
TERMTHDACT          ==> DUMP       QUIET,MSG,TRACE,DUMP,UAONLY,UADUMP,UAIMM,UATRACE

WHEN FINISHED, PRESS ENTER.
```

Figure 213. CLER run-time options transaction

### Notes:

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 you need to run CLER are in the CEE group, and 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 *OS/390 Language Environment Debugging Guide and Run-Time Messages*.



---

## Chapter 23. CMAC—messages and codes display

Use CMAC to give you individual message information as provided in the *CICS Messages and Codes* manual.

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* manual.

### Notes:

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 214 on page 560.

```
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 DFHxyyyy, Where xx is the Component ID.

Message Number => (for example, 1060, 5718, or Abend Code
                  such as ASRA, etc.)

F3=Exit to CICS
```

Figure 214. CMAC transaction: initial screen

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 215. 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 216. 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 24. 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 on page "CMSG command" on page 568.

Full details of the options are given under "CMSG command options" on page 568.

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.

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:

```
MSG 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 MSG 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.
- MSG as your first 4 characters, followed by end-of-data (EOD). You get the response:  
NO INPUT - REENTER.
- MSGx 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 29, “CSPG—page retrieval,” on page 609.

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 page “CMSG command options, ROUTE” on page 572), 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.

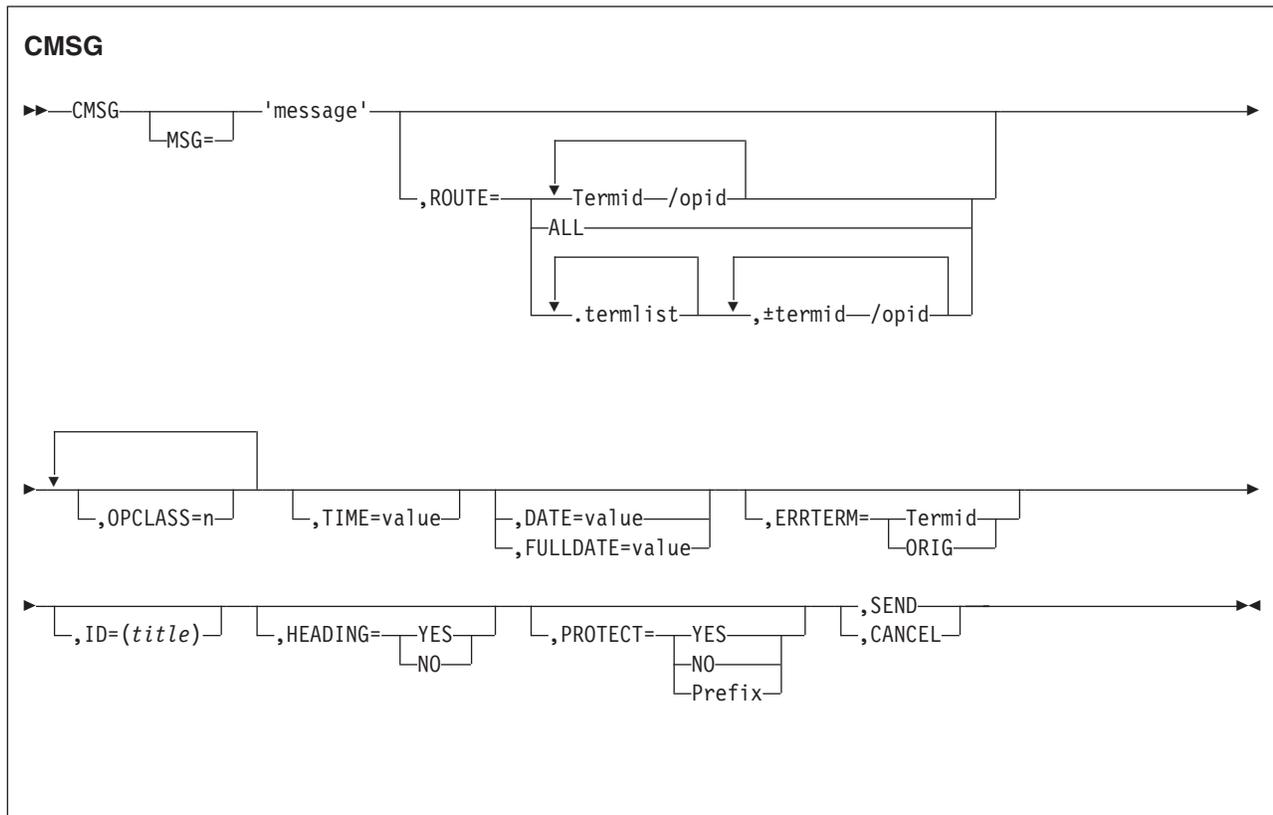
## IBM 3767 Communication Terminal

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.

## CMMSG command



## CMMSG 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<sup>3</sup> date is in the year 1997, January 31 1997 could be specified as 97.031,

3. 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.

01/31/97, or 01/31. In this last case, the year of the current system<sup>3</sup> 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/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<sup>3</sup> 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 on page “ CMSG command options, SEND” on page 575. 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<sup>3</sup> 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.

#### Notes:

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<sup>3</sup> 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<sup>4</sup> 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<sup>4</sup> 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<sup>4</sup> 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 on page " CMSG command options, SEND" on page 575. 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.

---

4. 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 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<sup>4</sup> 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<sup>4</sup> 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 29, "CSPG—page retrieval," on page 609 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 577.

### 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 `DFHTLTL2`. 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

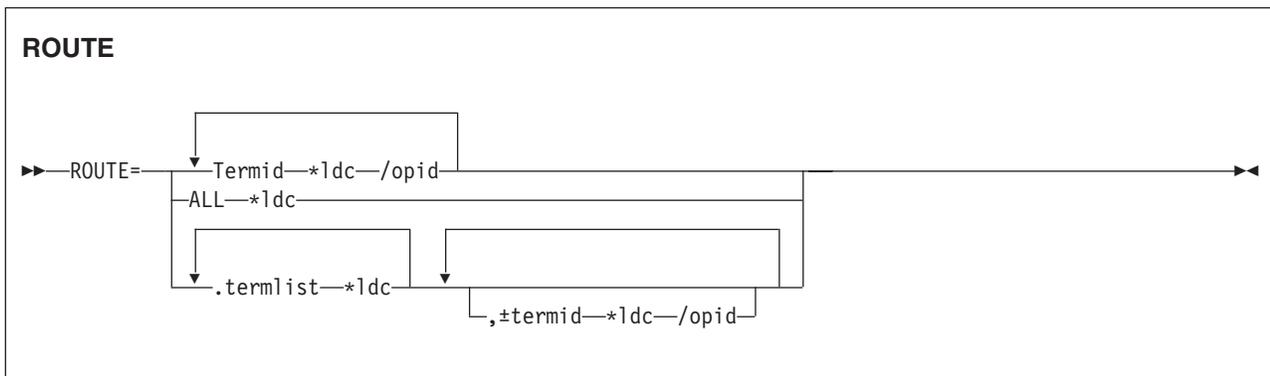
Some types of logical unit (LU) can be used to get access to more than one resource within a subsystem. For example, data sent to a 3601 LU may be intended for an IBM 3604 Keyboard Display, an IBM 3618 Administrative Line Printer, or some other element of the IBM 3600 Finance Communication System. The 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:

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.”

**+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**

These examples assume the following:

- Terminal identifiers T36n and T37n are 3600 logical units and 3770 batch logical units respectively; all others are start-stop or BSC terminals.
- LDC mnemonics DS and LP have been defined for the system and T361, T362, and T363 during resource definition.
- LDC mnemonic P1 has been defined for the system and T371, T372, and T373 during resource definition.

You can also use the TYPETERM LDCLIST attribute of RDO to define these mnemonics instead of the DFHTCT TYPE=TERMINAL macro shown.

- Terminal list table DFHTLTL3 contains entries for:

```
T361*DS
T362*DS/OP1
T362*DS/OP2
```

T363  
T371\*P1  
T372\*P1/OP1  
T372\*P1/OP2  
T373  
T40.

- Terminal list table DFHTLTL4 contains entries for:

T361\*DS  
T362\*DS/OP1  
T362\*DS/OP2  
T363  
T40.

**1. R=T361\*DS**

Route message to terminal identifier T361 qualified by LDC mnemonic DS.

**2. R=(T361\*DS,T362\*DS/OP1,T363,T371\*P1, T372\*P1/OP1,T373,T40)**

Route message to:

T361 and T362 qualified by LDC mnemonic DS  
T363, using the default LDC  
T371 and T372 qualified by LDC mnemonic P1  
T373 console keyboard/printer  
T40, which is a start-stop or BSC terminal.

T362 and T372 require that an operator with identification OP1 be signed on before the message can be sent.

**Note:** The default LDC mnemonic for T363 must indicate the same device type as LDC mnemonic DS.

**3. R=ALL\*LP**

Route message to all terminals (3600, 3770 batch and interactive logical units, start-stop, and BSC) with all 3600 destinations qualified by mnemonic LP. LP is ignored for start-stop and BSC destinations.

**4. R=.L3**

Use the terminal list table DFHTLTL3 for message destinations. (This is the same as example 2, plus T362\*DS/OP2 and T372\*P1/OP2.)

**5. R=(.L3,-T362\*DS/OP2,-T372\*P1/OP2)**

This is the same as example 4, but deletes T362\*DS/OP2 and T372\*P1/OP2, so is the same result as example 2.

**6. R=(.L3,-T362\*DS)**

This is the same as example 4, but deletes all entries for T362\*DS (with or without opids). The result is T361\*DS, T363, all T37n terminals, and T40.

**7. R=(.L3,-T362)**

This is the same as example 6. -T362 deletes all entries for T362.

**8. R=.L4\*LP**

LDC mnemonic LP qualifies (overrides) all entries in DFHTLTL4. Resulting destinations are:

T361\*LP T362\*LP/OP1 T362\*LP/OP2 T363\*LP T40\*LP

**Note:** The LP mnemonic has no effect on the start-stop or BSC terminal T40.

**9. R=(.L4\*LP,-T362\*DS)**

The -T362\*DS causes no action, because the TLT destinations are qualified by LDC mnemonic LP before the additions or deletes are processed, thus causing no matching entry to delete.

**10. R=(.L4,+T363\*LP)**

Cause error message 'INVALID LDC AT T363\*LP' to be generated. LDC mnemonic LP has a different device type from LDC mnemonic DS (first 3600 destination encountered is T361\*DS). All 3600 destinations for one message must indicate the same device type. All 3770 batch logical unit destinations for one message must indicate the same device type.

**11. R=/OP2**

Route message to the first terminal or logical unit found in the CICS terminal definition with operator identifier OP2 currently signed on. If OP2 is signed on to T362, the resulting destination is T362/OP2 with the default LDC mnemonic for logical unit T362. The default is DS because it is the first LDC mnemonic defined for T362. The resulting destination is T362\*DS/OP2.

---

## 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 565.

**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.

**Notes:**

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 xxxxx**

A HEADNG option may only be specified as follows:

```

HEADNG
HEADNG=YES
HEADNG=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 xxxxxx**

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 xxxxxx**

A terminal list table suffix must be 1 or 2 characters preceded by a period.

**INVLD OPERATR ID AT xxxxxx**

An operator identifier must be 1, 2, or 3 characters preceded by a slash.

**INVLD SEPARATOR AT xxxxxx**

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 xxxxx**

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 xxxxxx**

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.

---

## CMMSG examples

Here are some typical examples of the CMMSG transaction.

In these examples, # represents a new-line character.

**1. CMMSG R=ALL,'GOOD MORNING#TODAY IS FRIDAY FEB 1',S**

This sends a good morning message to all terminals.

**2. CMMSG '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. CMMSG '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. CMMSG '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. CMMSG 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 terminal.

**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 25. CPIA—unit of work resynchronization**

#

To avoid a warm restart, use the CPIA transaction to resynchronize Web Services Atomic Transaction (WS-AT) units of work and to remove shunted units of work.

#

#

Before you run the CPIA transaction, you must have the D-type APAR PK60532 installed.

#

#

To run the CPIA transaction, enter the CPIA command in the terminal.



---

## Chapter 26. 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.

**Note:** 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 592
- “CREC” on page 602

**Access to CREA:** By default, users who are authorised to use CEDA can use CREA.

---

### 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.

#### Notes:

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,

---

## 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.

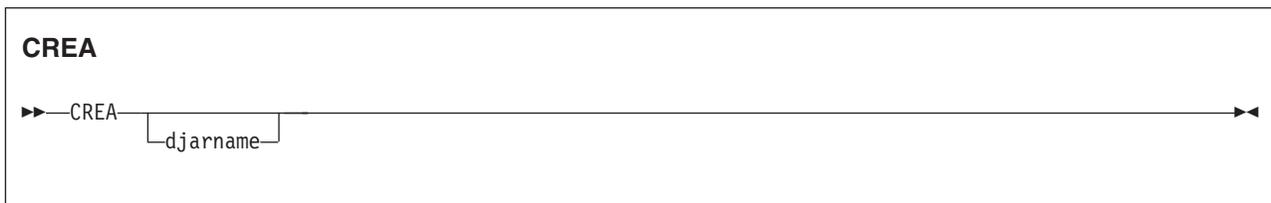
**Note:** Choose a suitable code page to connect to your CICS system. If you don't, you'll see the square bracket characters [ and ] displaying strangely. One suitable code page is 1047 (United States English).

## Starting CREA

Start the transaction by entering CREA (optionally followed by the name of a DJAR definition), on the command line of the screen. 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 218 on page 594), 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 217 on page 593) and a message indicating the state of the DJAR, or indicating that the DJAR was not found, (as is appropriate).

### Command syntax



### 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 217. CREA transaction: initial screen

If you enter CREA with a valid DJAR name as a parameter, the initial screen (Figure 217) is not presented to you. Instead you are taken directly to the TranID Specification screen (Figure 218 on page 594).

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 217.

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 218 on page 594).

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 218. CREA transaction: TranID Specification screen

The example given in Figure 218 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 REQUESTMODEL 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 220 on page 597).

### 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 219 on page 597, 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 219 on page 597. 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 219. CREA transaction: TranID Specification screen

## REQUESTMODEL Specification

The REQUESTMODEL Specification screen (Figure 220) 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 220. 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 599.

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 overwrite 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 221 is displayed, summarising 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 221. 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 217 on page 593).

---

## 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.

**Access to CREC:** By default, users who are authorised 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 218 on page 594) 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 220 on page 597), and the REQUESTMODEL results screen (Figure 221 on page 600) are not available.

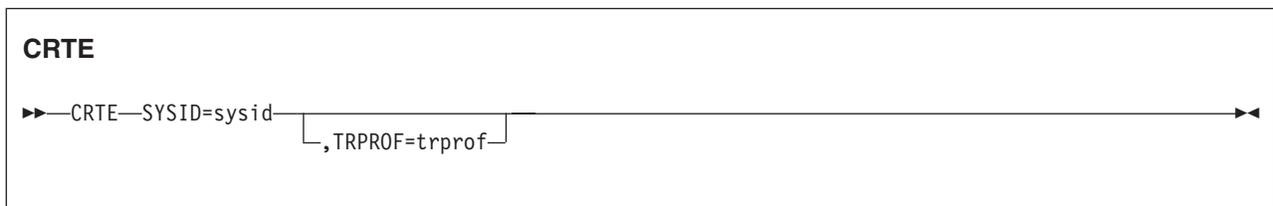
## Chapter 27. CRTE—remote transactions

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.

The syntax of the routing transaction is:



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 to be used for the session that carries the intersystem flows, if that session uses VTAM. If VTAM is used and 'trprof' 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 has ended, you get the message:

```
ROUTING SESSION TO SYSTEM system-id HAS BEEN TERMINATED
```

**Notes:**

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 28. 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).

You start the transaction by entering CSFE on the command line of the screen, as follows:

```
CSFE
```

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.



### OFF

switches off the system spooling interface trace and the diagnostic facility.

### ON

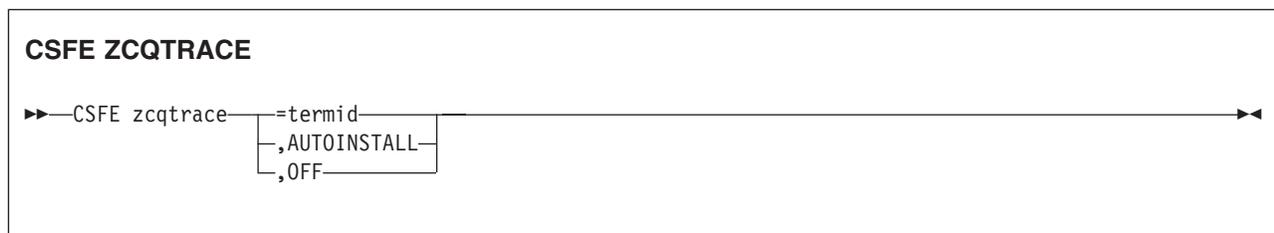
specifies that additional trace entries of system type are provided for the system spooling interface.

### 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.



### ,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.

---

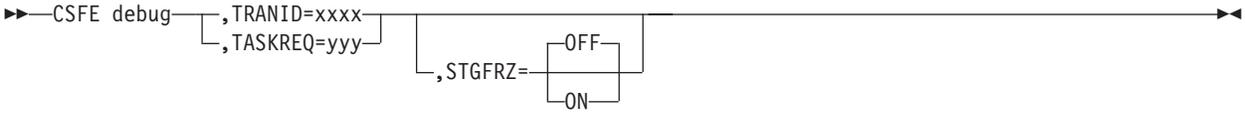
## Using CSFE debug functions

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 may have occurred during processing.

## CSFE DEBUG TRANID



You must code the options in the order shown in the syntax display above.

### 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



### 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.

For information on the CSFE options and their actions, see the *CICS Problem Determination Guide*.

**NONE**

deactivates the user storage checking function.

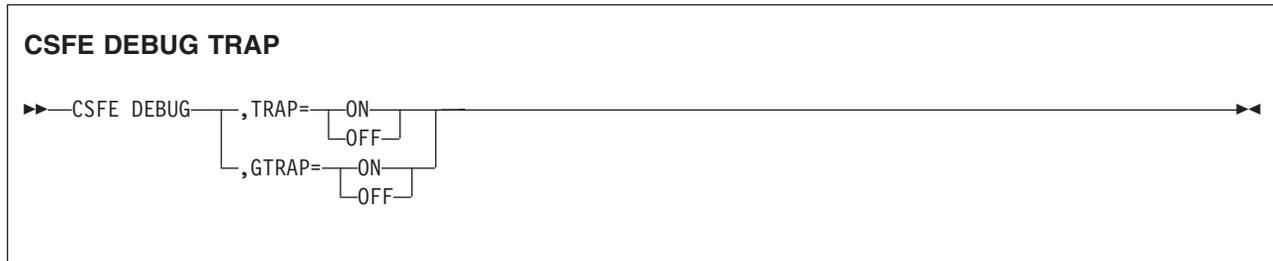
**CHKSTRM={CURRENT|NONE}**

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.



**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.

**TRAP= ON or off**

specifies that the global/trap exit routine is to be activated or deactivated.

---

## Chapter 29. 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.

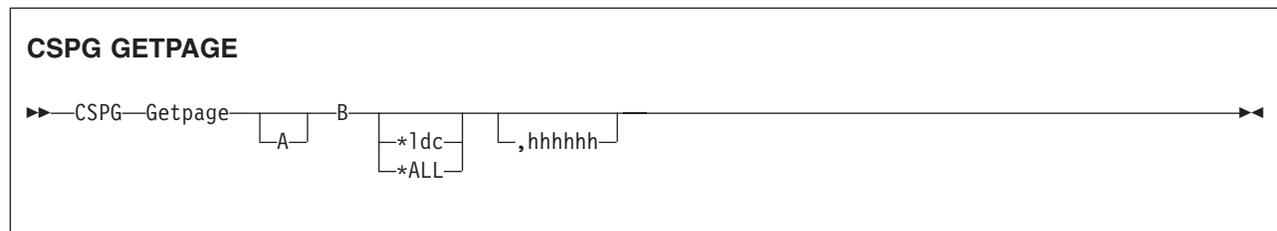
You select the above functions by entering a unique identification code for each function. This code is 1–7 characters long, and is defined by the PGRET, PGCHAIN, PGPURGE, and PGCOPY system initialization parameters.

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 555).



#### 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 612.

- 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 616.

- X** enter data that is not recognized as a paging command.

This is discussed further under “Single keystroke retrieval (SKR)” on page 611.

**\*1dc**

Is the LDC for the device to which you want to send the message. (See Page retrieval for logical devices.)

**\*ALL**

A special case of \*1dc.

**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 612).

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*Idc`-resets the specified logical device to autopage and delivers the pages for that \*Idc. 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

This command displays the message identifiers and titles of logical messages queued for this terminal.

```
getpageQ
```

### Getpage

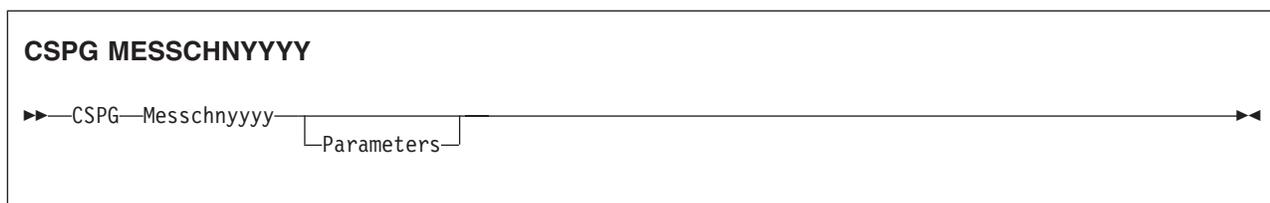
The user-defined identification code for the retrieval transaction, and is 1–7 characters long. The PGRET system initialization parameter defines this code.

- Q** (Query). The identifiers of all logical messages destined for this terminal are displayed. If the message is security protected (that is, if it has been routed to a specific operator ID or operator class), its identifier is displayed only if the operator ID or class for the operator signed on matches that in the message. The identifier consists of a 6-digit hexadecimal number, a 2-character temporary-storage prefix, and optionally, a title. The 6-digit hexadecimal number can be used as the hhhhhh option in the page retrieval command (see "Retrieve a page" on page 609) to retrieve pages for this display.

---

## 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:



**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 615. 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).

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...

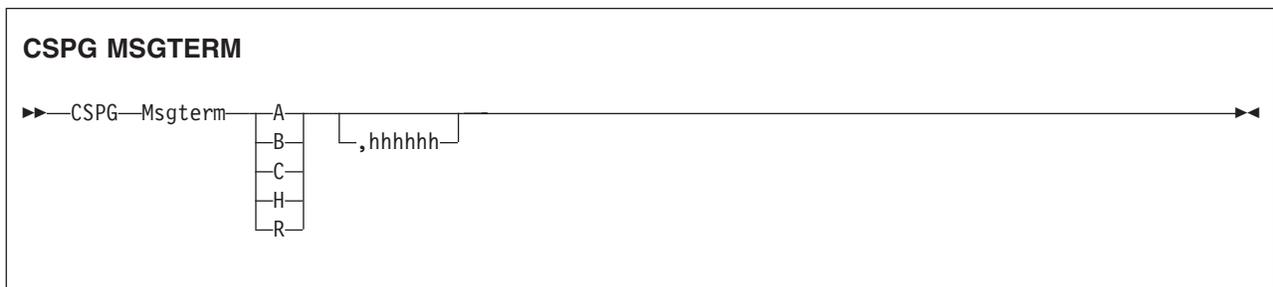
You can chain a new transaction only while you are viewing the last logical level. If you have four levels (as in the above example), view the summary page (level one) and then enter C/PREC 3642; the new transaction becomes level two and all other chained messages below that are erased (assuming that 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:



### 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 (hhhhh) 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 by entering the page-copy transaction identifier and the identifier of the terminal that is to receive the copy, as follows:

```
msgcopytttt
```

**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 24, “CMSG—message switching,” on page 563. 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 30. 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

<p><b>CWTO</b></p> <p>▶ CWTO—Message ▶</p>
--

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



---

## Chapter 31. 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*.



## 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, and which are therefore described in this manual, are marked by an asterisk (\*). The remainder therefore have no operator interface.

**Tranid** is the transaction identifier. **Program** indicates the program (if there is one) that is specified in the supplied transaction definition, and **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*.

Tranid	Program	Group	Description
CADP*	DFHDPLU	DFHDP	Application debugging profile manager
CAQP	DFHAQP		Asynchronous Queue Purge Program
CATA	DFHZATA	DFHSPI	Autoinstall terminal
CATD	DFHZATD	DFHSPI	Delete autoinstalled terminal
CATP	DFHATP		Asynchronous Transaction Control Program
CATR	DFHZATR	DFHSPI	Restart autoinstalled terminal
CATS	DFHZATS		
CAUT	DFHSTSP		Automatic System Statistics Program
CBAM*	DFHECBAM	DFHOPER	BTS browser
CBRA	DFHSTSP		Bridge facility allocation
CCIN	DFHZCN1	DFHCLNT	CICS Client
CCMF	DFHCCMF		Monitoring ATI Program
CCMS	EFHCCMS		CMS Control Transaction
CCRL*	DFHSOCRL	DFHOPER	CICS Certificate Revocation List transaction
CCU2	EFHUCMS2		CMS VSAM Conversion utility
CDBC*	DFHDBME	DFHDBCTL	DBCTL connect/disconnect
CDBD	DFHDBDI	DFHDBCTL	DBCTL disable
CDBF	DFHD2CM3	DFHDB2	CICS/DB2 shutdown force transaction
CDBI*	DFHDBIQ	DFHDBCTL	DBCTL inquiry
CDBM*	DFHDBMP	DFHDBCTL	DBCTL issue commands
CDBN	DFHDBCON	DFHDBCTL	DBCTL issue commands
CDBO	DFHDBCT	DFHDBCTL	DBCTL control
CDBQ	DFHD2CM2	DFHDB2	CICS DB2 attachment facility shutdown quiesce transaction
CDBT	DFHDBDSC	DFHDBCTL	DBCTL disconnection
CDFS	DFHDFST	DFHISC	CICS Deferred Start for non-term starts
CDST	DFHMIRS		CICS Dynamic routing of non-term starts
CDTS	DFHZATS	DFHSPI	
CEBR*	DFHEDFBR	DFHEDF	Temporary-storage browse
CEBT*	—	—	Master terminal, alternate CICS
CECI*	DFHECIP	DFHINTER	Command level interpreter
CECS*	DFHECSP	DFHINTER	Command level interpreter

<b>Tranid</b>	<b>Program</b>	<b>Group</b>	<b>Description</b>
CEDA*	DFHEDAP	DFHSPI	Resource definition online (RDO)
CEDB*	DFHEDAP	DFHSPI	Resource definition online (RDO)
CEDC*	DFHEDAP	DFHSPI	Resource definition online (RDO)
CEDF*	DFHEDFP	DFHEDF	Execution diagnostic facility
CEDX*	DFHEDFP	DFHEDF	Execution diagnostic facility
CEGN	DFHCEGN	DFHSIGN	Good night transaction stub
CEHP	DFHCHS	DFHISC	CICS/VM™ request handler
CEHS	DFHCHS	DFHISC	CICS/VM request handler
CEID	DFHCEID		UUID Timeout Transaction
CEJR			Emergency use
CEKL*			
CEMS	DFHEMSP		Spooling Master Terminal Transaction
CEMT*	DFHEMTP	DFHOPER	Master terminal
CEOS	DFHEOSP		Spooling Terminal Operator Transaction
CEOT*	DFHEOTP	DFHOPER	Terminal status
CEPC	DFHPSCP		Printer Spooling Print Control
CEPE	DFHPSEP		Printer Spooling Report Deletion
CEPQ	DFHP SOP		Print Intra-Partition TD Queue
CEPW	DFHP SOP		Printer Spooling Writer
CESC*	DFHCESC	DFHSIGN	Terminal timeout
CESD	DFHCESD	DFHSDAP	Shutdown assist
CESF*	DFHSFP	DFHSIGN	Sign off
CESN*	DFH SNP	DFHSIGN	Sign on
CEST*	DFHESTP	DFHOPER	Supervisory terminal
CETR*	DFHCETRA	DFHOPER	Trace
CEX2	DFHDB2EX2	DFHDB2	CICS DB2 attachment facility service transaction
CFCL	DFHFCDL	-	CFDT Load
CFOR	DFHFCQT	-	RLS Quiesce
CFQS	DFHFCQT	-	RLS Quiesce
CFTI			Flowmark for MVS
CFTL	DFHDTLX	-	Shared DT load
CFTM			Boeblingen Banking Project
CFTS	DFHZATS	DFHSPI	Remote mass flag
CFUP			Boeblingen Banking Project
CGRP	-	-	Persistent sessions (See note)
CHLP	DFHCMAC	DFHCMAC	Alias for CMAC
CIDP*	DFHDPIN	DFHDP	Inactivate debugging profiles
CIEP	DFHI EP	DFHIPECI	ECl over TCP/IP
CIND*	DFHINDT	DFHINDT	CICS in-doubt testing tool
CFUP			Boeblingen Banking Project
CINS			
CIOD	DFHIIOPA	DFHIIOP	Default IIOp interface, started by CIOR
CIOF	DFHIIOPA	DFHIIOP	CICS Generic factory started by CIOR
CIOR	DFHIIOP	DFHIIOP	CICS IIOp interface started by SO_Domain
CIRB			SQL/DS Tranid
CIRD			SQL/DS Tranid
CIRP	DFHII RP	DFHIIOP	Default CICS IIOp request processor transaction

<b>Tranid</b>	<b>Program</b>	<b>Group</b>	<b>Description</b>
CIRR	DFHIIRRS	DFHIIOP	Default CICS IIOF request recover transaction
CITS	DFHZATS	DFHSPI	Remote install
CJMJ	DFHSJJM	DFHJAVA	Master JVM
CJTR	DFHOTR		OTS Resync
CLEP			LE runtime options
CLER			LE runtime options
CLQ2	DFHLUP	DFHISC	OUTBOUND RESYNC FOR APPC AND MRO
CLR1	DFHZLS1	DFHISCT	Inbound CNOS for APPC parallel sessions connections
CLR2	DFHLUP	DFHISC	INBOUND RESYNC FOR MRO
CLS1	DFHZLS1	DFHISC	LU services model
CLS2	DFHLUP	DFHISC	LU services model
CLS3	DFHCLS3	DFHISC	LU services model
CLS4	DFHCLS4	DFHISC	LU services model
CMAC	DFHCMAC		Messages utility
CMPX	DFHMPX	DFHISC	Local queuing shipper
CMSG*	DFHMSP	DFHMSWIT	Message switching
CMTS	DFHZATS	DFHSPI	Remote mass delete
COVR	DFHZCOVR	-	CICS open VTAM retry
CPIS	DFHPIR	DFHPIPE	WS-AT transaction that is attached when resynchronization is required.
CPLT	-	-	PLT programs
CPMI	DFHMIRS	DFHISC	Synchronization level 1 mirror
CQPI	DFHCLS5	DFHISC	Connection quiesce transaction inbound
CQPO	DFHCLS5	DFHISC	Connection quiesce transaction outbound
CQRY	DFHQRY	DFHSTAND	ATI query support
CRDR	DFHRD1		Asynchronous Transaction Processing in Put Routine 1
CREA*	DFHADDRM	DFHADST	Create REQUESTMODELS for enterprise beans
CREC*	DFHADDRM	DFHADST	Create REQUESTMODELS for enterprise beans (read only)
CRMD	DFHZATMD	DFHSPI	Remote terminal delete
CRMF	DFHZATMF	DFHSPI	Remote terminal flag for delete
CRPA	DFHRPAS	DFHRPC	ONC RPC-supplied transaction
CRPC	DFHRPC00	DFHRPC	ONC RPC-supplied transaction
CRPM	DFHRPHS	DFHRPC	ONC RPC-supplied transaction
CRSQ	DFHCRQ	DFHISC	ATI purge
CRSR	DFHCRS	DFHISC	Remote scheduler
CRSY	DFHRMSY	DFHRMI	Resource manager resynchronization
CRTE*	DFHRTE	DFHISC	Transaction routing
CRTP	DFHZRTP	DFHPSSGN	Persistent sessions sign on retention.
CRTX	-	DFHISC	Dynamic transaction routing transaction definition
CSAC	DFHACP	DFHSTAND	Abnormal condition
CSCY	DFHCPY	DFHVTAMP	3270 print support
CSFE*	DFHFEP	DFHFE	Terminal test, trace, storage
CSFR	DFHFCDR	-	RLS cleanup
CSFU	DFHFUCU	DFHOPCLS	File open utility
CSGM	DFHGMM	DFHVTAM	"Good-morning" signon
CSHA	-	-	Scheduler services

#

Tranid	Program	Group	Description
CSHQ	-	-	Scheduler services
CSHR	DFHMIRS	DFHISC	SCHEDULER SERVICES REMOTE ROUTING
CSIR 160	DFHCRR		Inter-region Session Recovery Program
CSJC	DFHJC BSP	DFHCOMP3	Journal control bootstrap
CSKD	DISABLE		Disable TCP/IP interface adapter
CSKE	ENABLE		Enable TCP/IP interface adapter
CSKL	LISTENER		TCP/IP Listener
CSKP	DFHAKP	DFHCOMP5	Activity keypoint
CSMI	DFHMIRS	DFHISC	Mirror transaction
CSMT	DFHMTPA		Master Terminal Module A
CSM1	DFHMIRS	DFHISC	SYMSMSG model
CSM2	DFHMIRS	DFHISC	Scheduler model
CSM3	DFHMIRS	DFHISC	Queue model
CSM5	DFHMIRS	DFHISC	DL/I model
CSNC	DFHCRNP	DFHISC	IRC connection manager
CSNE	DFHZNAC	DFHVTAM	VTAM node abnormal
CSOL	DFHSOL	-	TCPIP listener
CSOT	DFHMTPA		Master Terminal Module A
CSPG*	DFHTPR	DFHBMS	Terminal paging
CSPK	DFHPRK	DFHVTAMP	3270 print support
CSPP	DFHP3270	DFHHARDC	3270 print support
CSPQ	DFHTPQ	DFHBMS	Terminal page cleanup
CSPS	DFHTPS	DFHBMS	Terminal page scheduling
CSQC	DFHLGQC	DFHLGQC	CICS quiesce after system log failure
CSRK	DFHRKB	DFHVTAMP	3270 print support
CSRS	DFHZRSP	DFHRSEND	3614 message resynchronization
CSSC	DFHCSSC>		Sign off inactive terminal
CSSF	DFHRTC	DFHISC	CRTE cancel remote transaction
CSSN	DFH SNP		Sign On Program
CSST	DFHMTPA		Master Terminal Module A
CSSY	—	—	(See note)
CSTA	DFHTAJP		Time of Day Adjustment Program
CSTB			
CSTE	DFHTACP	DFHSTAND	Terminal abnormal
CSTP	—	—	(See note)
CSTT	DFHSTKC		Supervisory Statistics Program
CSZI	DFHSZRMP	DFHFPEI	Front end programming interface (FEPI), only active if FEPI installed.
CTIN	DFHZCT1	DFHCLNT	CICS Client
CTSD	DFHTSDQ	-	Temporary storage delete
CVMI	DFHMIRS	DFHISC	Synchronization level 1 mirror
CVST	DFHVAP		Subtask Monitor Program
CWBA	DFHWBA	DFHWEB	CICS web interface alias transaction
CWBC	DFHWBC00	DFHWEB	CICS web interface connection manager
CWBG	DFHWBGB	DFHWEB	CICS web interface garbage collection
CWBM	DFHWBM	DFHWEB	CICS web interface server controller
CWTO*	DFHCWTO	DFHCONS	Write to console operator
CWTR	DFHWT1		Asynchronous Transaction Output Phase 1
CWXN	DFHWBXN	DFHWEB	CICS Web support attach transaction

|  
|

<b>Tranid</b>	<b>Program</b>	<b>Group</b>	<b>Description</b>
CWXU	DFHWBXN	DFHWEB	CICS Web support USER protocol attach transaction
CXCU	DFHCXCU	DFHSTAND	XRF catchup transaction
CXRE	DFHZXRE	DFHSTAND	Autoconnect processing
CXRT	DFHCRT	DFHISC	Transaction routing relay

**Note:** Transactions CGRP, CSSY, CSTP, and CSXM are transaction names that are used by some CICS tasks.



---

## 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 1:

*Memo to Licensees*, GI10-2559  
*CICS Transaction Server for z/OS Program Directory*, GI10-2586  
*CICS Transaction Server for z/OS Release Guide*, GC34-6421  
*CICS Transaction Server for z/OS Installation Guide*, GC34-6426  
*CICS Transaction Server for z/OS Licensed Program Specification*, GC34-6608

You can order further copies of the following books in the entitlement set, using the order number quoted above:

*CICS Transaction Server for z/OS Release Guide*  
*CICS Transaction Server for z/OS Installation Guide*  
*CICS Transaction Server for z/OS Licensed Program Specification*

### PDF-only books

The following books are available in the CICS Information Center as Adobe Portable Document Format (PDF) files:

#### CICS books for CICS Transaction Server for z/OS

##### General

*CICS Transaction Server for z/OS Program Directory*, GI10-2586  
*CICS Transaction Server for z/OS Release Guide*, GC34-6421  
*CICS Transaction Server for z/OS Migration from CICS TS Version 2.3*, GC34-6425

*CICS Transaction Server for z/OS Migration from CICS TS Version 1.3,*  
GC34-6423

*CICS Transaction Server for z/OS Migration from CICS TS Version 2.2,*  
GC34-6424

*CICS Transaction Server for z/OS Installation Guide,* GC34-6426

#### **Administration**

*CICS System Definition Guide,* SC34-6428

*CICS Customization Guide,* SC34-6429

*CICS Resource Definition Guide,* SC34-6430

*CICS Operations and Utilities Guide,* SC34-6431

*CICS Supplied Transactions,* SC34-6432

#### **Programming**

*CICS Application Programming Guide,* SC34-6433

*CICS Application Programming Reference,* SC34-6434

*CICS System Programming Reference,* SC34-6435

*CICS Front End Programming Interface User's Guide,* SC34-6436

*CICS C++ OO Class Libraries,* SC34-6437

*CICS Distributed Transaction Programming Guide,* SC34-6438

*CICS Business Transaction Services,* SC34-6439

*Java Applications in CICS,* SC34-6440

*JCICS Class Reference,* SC34-6001

#### **Diagnosis**

*CICS Problem Determination Guide,* SC34-6441

*CICS Messages and Codes,* GC34-6442

*CICS Diagnosis Reference,* GC34-6899

*CICS Data Areas,* GC34-6902

*CICS Trace Entries,* SC34-6443

*CICS Supplementary Data Areas,* GC34-6905

#### **Communication**

*CICS Intercommunication Guide,* SC34-6448

*CICS External Interfaces Guide,* SC34-6449

*CICS Internet Guide,* SC34-6450

#### **Special topics**

*CICS Recovery and Restart Guide,* SC34-6451

*CICS Performance Guide,* SC34-6452

*CICS IMS Database Control Guide,* SC34-6453

*CICS RACF Security Guide,* SC34-6454

*CICS Shared Data Tables Guide,* SC34-6455

*CICS DB2 Guide,* SC34-6457

*CICS Debugging Tools Interfaces Reference,* GC34-6908

### **CICSplex SM books for CICS Transaction Server for z/OS**

#### **General**

*CICSplex SM Concepts and Planning,* SC34-6459

*CICSplex SM User Interface Guide,* SC34-6460

*CICSplex SM Web User Interface Guide,* SC34-6461

#### **Administration and Management**

*CICSplex SM Administration,* SC34-6462

*CICSplex SM Operations Views Reference,* SC34-6463

*CICSplex SM Monitor Views Reference,* SC34-6464

*CICSplex SM Managing Workloads,* SC34-6465

*CICSplex SM Managing Resource Usage,* SC34-6466

*CICSplex SM Managing Business Applications,* SC34-6467

#### **Programming**

*CICSplex SM Application Programming Guide,* SC34-6468

*CICSplex SM Application Programming Reference,* SC34-6469

## Diagnosis

*CICSplex SM Resource Tables Reference*, SC34-6470  
*CICSplex SM Messages and Codes*, GC34-6471  
*CICSplex SM Problem Determination*, GC34-6472

## CICS family books

### Communication

*CICS Family: Interproduct Communication*, SC34-6473  
*CICS Family: Communicating from CICS on System/390*, SC34-6474

### Licensed publications

The following licensed publications are not included in the unlicensed version of the Information Center:

*CICS Diagnosis Reference*, GC34-6899  
*CICS Data Areas*, GC34-6902  
*CICS Supplementary Data Areas*, GC34-6905  
*CICS Debugging Tools Interfaces Reference*, GC34-6908

---

## Other CICS books

The following publications contain further information about CICS, but are not provided as part of CICS Transaction Server for z/OS, Version 3 Release 1.

<i>Designing and Programming CICS Applications</i>	SR23-9692
<i>CICS Application Migration Aid Guide</i>	SC33-0768
<i>CICS Family: API Structure</i>	SC33-1007
<i>CICS Family: Client/Server Programming</i>	SC33-1435
<i>CICS Transaction Gateway for z/OS Administration</i>	SC34-5528
<i>CICS Family: General Information</i>	GC33-0155
<i>CICS 4.1 Sample Applications Guide</i>	SC33-1173
<i>CICS/ESA 3.3 XRF Guide</i>	SC33-0661

---

## 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.



---

# Index

## Special characters

- and CEMT 135
- ? and CEMT 135
- (value)
  - CEMT INQUIRE JMODEL 251
  - CEMT INQUIRE JOURNALNAME 253
  - CEMT SET PROCESSTYPE 478
- \* and CEMT 139
- \*ALL
  - CSPG 610
- \*LDC
  - CMSG 577
- TERMINID\*LDC/OPID
  - CMSG 577
- + in CEMT syntax 139
- + in scrolling under CEMT 141
- +HHMM
  - CMSG 585
- +M
  - CMSG 586
- +MM
  - CMSG 585
- +TERMINID\*LDC/OPID
  - CMSG 577

## Numerics

- 3270 interface
  - application debugging profile manager 37

## A

- abbreviation of keywords
  - CEMT transaction 139
- abbreviations for keywords 3
- ABEND
  - CEMT INQUIRE DB2CONN 189, 192
  - CEMT INQUIRE DB2ENTRY 195
  - CEMT SET DB2CONN 429
  - CEMT SET DB2ENTRY 437
- access method control block (ACB) 380, 515
- ACCESSMETHOD
  - CEMT INQUIRE CONNECTION 175
  - CEMT INQUIRE DSNAME 216
  - CEMT INQUIRE FILE 237
- ACCOUNTREC
  - CEMT INQUIRE DB2CONN 186
  - CEMT INQUIRE DB2ENTRY 194
  - CEMT SET DB2CONN 426
  - CEMT SET DB2ENTRY 435
- ACQ
  - CEOT 522
- ACQSTATUS
  - CEMT INQUIRE MODENAME 264
- ACQUIRED
  - CEMT INQUIRE CONNECTION 176
  - CEMT INQUIRE MODENAME 264

- ACQUIRED (*continued*)
  - CEMT INQUIRE NETNAME 273
  - CEMT INQUIRE TERMINAL 337
  - CEMT SET CONNECTION 418
  - CEMT SET MODENAME 468
  - CEMT SET NETNAME 473
  - CEMT SET TERMINAL 503
- acquiring VTAM terminals 473, 503
- ACTION
  - CEMT INQUIRE DSNAME 216
  - CEMT INQUIRE SYDUMPCODE 303
  - CEMT INQUIRE TRDUMPCODE 345
  - CEMT SET TSQUEUE 510
- activating debugging profiles
  - with the 3270 interface 40
  - with the web interface 21
- ACTIVE
  - CEMT INQUIRE ENQ 228
  - CEMT INQUIRE MODENAME 264
  - CEMT INQUIRE SYSTEM 310
  - CEMT INQUIRE TCLASS 318
  - CEMT INQUIRE UOW 360
  - CEMT INQUIRE UOWENQ 369
- ACTIVITY
  - CEMT INQUIRE TASK 313
  - CEMT SET PROCESSTYPE 477
- ACTIVITYID
  - CEMT INQUIRE TASK 313
- ACTJVMTCBS
  - CEMT INQUIRE DISPATCHER 202
- ACTOPENTCBS
  - CEMT INQUIRE DISPATCHER 202
- ACTSOCKETS
  - CEMT INQUIRE TCPIP 319
- ACTSSLTCBS
  - CEMT INQUIRE DISPATCHER 203
- ACTXPTCBS
  - CEMT INQUIRE DISPATCHER 203
- ADD
  - CEMT INQUIRE SYDUMPCODE 303
  - CEMT INQUIRE TRDUMPCODE 345
  - CEMT SET SYDUMPCODE 487
  - CEMT SET TRDUMPCODE 508
- ADDABLE
  - CEMT INQUIRE FILE 237
  - CEMT SET FILE 454
- ADDREXECENQ
  - CEMT INQUIRE ENQ 228
  - CEMT INQUIRE UOWENQ 369
- ADDSTATUS
  - CEMT INQUIRE FILE 237
- AFFINITY
  - CEMT INQUIRE CONNECTION 175
- AFTER
  - CEDA 110
- AGE
  - CEMT INQUIRE JVM 257
  - CEMT INQUIRE UOW 358

## AGING

CEMT INQUIRE DISPATCHER 203  
CEMT INQUIRE SYSTEM 305  
CEMT SET DISPATCHER 442  
CEMT SET SYSTEM 489

## AIBRIDGE

CEMT INQUIRE AUTOINSTALL 156  
CEMT SET AUTOINSTALL 412

## AKP

CEMT INQUIRE SYSTEM 306  
CEMT SET SYSTEM 490

## ALL

CEBT INQUIRE AUXTRACE 80  
CEBT SET AUXTRACE 91  
CEDA 110  
CEMT INQUIRE AUTINSTMODEL 155  
CEMT INQUIRE AUXTRACE 160  
CEMT INQUIRE BEAN 162  
CEMT INQUIRE BRFACILITY 165  
CEMT INQUIRE CONNECTION 173  
CEMT INQUIRE CORBASERVER 183  
CEMT INQUIRE DB2CONN 186  
CEMT INQUIRE DB2ENTRY 193  
CEMT INQUIRE DB2TRAN 198  
CEMT INQUIRE DJAR 206  
CEMT INQUIRE DSNAME 214  
CEMT INQUIRE ENQMODEL 231  
CEMT INQUIRE FILE 235  
CEMT INQUIRE JMODEL 251  
CEMT INQUIRE JOURNALNAME 253  
CEMT INQUIRE MODENAME 263  
CEMT INQUIRE NETNAME 269  
CEMT INQUIRE PARTNER 275  
CEMT INQUIRE PIPELINE 277  
CEMT INQUIRE PROCESSTYPE 280  
CEMT INQUIRE PROFILE 282  
CEMT INQUIRE PROGRAM 283  
CEMT INQUIRE REQUESTMODEL 291  
CEMT INQUIRE STREAMNAME 299  
CEMT INQUIRE SYDUMPCODE 302  
CEMT INQUIRE TASK 311  
CEMT INQUIRE TCLASS 317  
CEMT INQUIRE TDQUEUE 326  
CEMT INQUIRE TERMINAL 332  
CEMT INQUIRE TRANSACTION 338  
CEMT INQUIRE TRDUMPCODE 344  
CEMT INQUIRE TSMODEL 347  
CEMT INQUIRE TSPPOOL 350  
CEMT INQUIRE TSQUEUE 352  
CEMT INQUIRE UOW 356  
CEMT INQUIRE UOWLINK 371  
CEMT INQUIRE URIMAP 375  
CEMT INQUIRE WEBSERVICE 383  
CEMT INQUIRE WORKREQUEST 388  
CEMT PERFORM STATISTICS 408  
CEMT SET AUXTRACE 414  
CEMT SET BRFACILITY 416  
CEMT SET CONNECTION 419  
CEMT SET CORBASERVER 424  
CEMT SET DSNAME 447  
CEMT SET FILE 455

## ALL (continued)

CEMT SET JOURNALNAME 463  
CEMT SET MODENAME 468  
CEMT SET NETNAME 473  
CEMT SET PIPELINE 476  
CEMT SET PROCESSTYPE 478  
CEMT SET PROGRAM 479  
CEMT SET SYDUMPCODE 487  
CEMT SET TASK 493  
CEMT SET TCLASS 495  
CEMT SET TDQUEUE 501  
CEMT SET TERMINAL 503  
CEMT SET TRANSACTION 506  
CEMT SET TRDUMPCODE 508  
CEMT SET TSQUEUE 510  
CEMT SET UOW 511  
CEMT SET UOWLINK 512  
CEMT SET WEBSERVICE 518  
CEMT SET WORKREQUEST 519  
CMSG 573  
    general statement 139  
ALL ROUTE ENTRIES DELETED  
    CMSG 580  
ALL\*LDC  
    CMSG 577  
ALLOCAGE  
    CEMT INQUIRE JVM 257  
ANALYZER  
    CEMT INQUIRE URIMAP 377  
ANALYZERSTAT  
    CEMT INQUIRE URIMAP 377  
ANY  
    CEMT INQUIRE PROGRAM 288  
    CEMT INQUIRE TRANSACTION 343  
APIST  
    CEMT INQUIRE PROGRAM 286  
APPC  
    CEMT INQUIRE CONNECTION 178  
    CEMT INQUIRE UOWLINK 373  
APPENDCRLF  
    CEMT INQUIRE DOCTEMPLATE 209  
application debugging profile manager 15  
    3270 interface 37  
        activating debugging profiles 40  
        changing debugging profiles 40  
        copying debugging profiles 41  
        Create Compiled Debugging Profile screen 42  
        Create Java Debugging Profile screen 47  
        creating debugging profiles 39  
        deleting debugging profiles 41  
        inactivating debugging profiles 40  
        List Debugging Profiles screen 37  
        Set Compiled Debugging Display Device  
            screen 54  
        setting the display device 53  
        View Compiled Debugging Profile screen 50  
        View Java Debugging Profile screen 51  
        viewing debugging profiles 40  
web interface 16  
    activating debugging profiles 21  
    changing debugging profiles 22

application debugging profile manager *(continued)*  
   web interface *(continued)*  
     configuring 16  
     copying debugging profiles 23  
     Create compiled profile page 24  
     Create Java profile page 28  
     creating debugging profiles 21  
     deleting debugging profiles 23  
     inactivating debugging profiles 22  
     List all profiles page 17  
     Set compiled display device page 34  
     setting the display device 33  
     View compiled profile page 31  
     View Java profile page 32  
     viewing debugging profiles 22  
 APPLNAME  
   CEMT INQUIRE MONITOR 266  
 APPLNAMEST  
   CEMT INQUIRE MONITOR 266  
 AS  
   CEDA 110  
 ASSEMBLER  
   CEMT INQUIRE PROGRAM 289  
 ASSERTED  
   CEMT INQUIRE CORBASERVER 183  
 ATI  
   CEMT INQUIRE NETNAME 270  
   CEMT INQUIRE TERMINAL 334  
   CEMT SET NETNAME 474  
   CEMT SET TERMINAL 503  
   CEOT 522  
 ATI (automatic transaction initiation) 330, 501  
 ATISTATUS  
   CEMT INQUIRE NETNAME 270  
   CEMT INQUIRE TERMINAL 333  
 AUDITLEVEL  
   CEMT INQUIRE PROCESSTYPE 281  
 AUDITLOG  
   CEMT INQUIRE PROCESSTYPE 281  
 AUT  
   CEOT 522  
 AUTHENTICATE 321  
 AUTHID  
   CEMT INQUIRE DB2CONN 187  
   CEMT INQUIRE DB2ENTRY 195  
   CEMT SET DB2CONN 427  
   CEMT SET DB2ENTRY 436  
 AUTHTYPE  
   CEMT INQUIRE DB2CONN 187  
   CEMT INQUIRE DB2ENTRY 195  
   CEMT SET DB2CONN 427  
   CEMT SET DB2ENTRY 436  
 AUTINSTMODEL  
   CEMT DISCARD 147  
   CEMT transaction 154  
 AUTOACTIVE  
   CEMT INQUIRE SYSTEM 308  
   CEMT SET SYSTEM 490  
 AUTOINACTIVE  
   CEMT INQUIRE SYSTEM 308  
   CEMT SET SYSTEM 490

AUTOINSTALL  
   CEMT INQUIRE AUTOINSTALL 157  
   CEMT PERFORM STATISTICS 408  
   CEMT SET transaction 412  
   CEMT transaction 156  
   CSFE ZCQTRACE 606  
 AUTOMATIC  
   CEBT SET TAKEOVER 97  
 automatic transaction initiation (ATI) 330, 501  
 AUTOPAGEABLE  
   CEMT INQUIRE NETNAME 272  
   CEMT INQUIRE TERMINAL 335  
   CEMT SET NETNAME 474  
   CEMT SET TERMINAL 503  
   CEOT 522  
 autopaging terminal 616  
 AUTOPUB  
   CEMT SET CORBASERVER 424  
 AUTOPUBLISH  
   CEMT INQUIRE CORBASERVER 183  
 AUTOSTARTST  
   CEMT INQUIRE CLASSCACHE 170  
 AUTOSWITCH  
   CEBT INQUIRE DUMP 82  
   CEBT SET DUMP 93  
   CEMT INQUIRE DUMPDS 224  
   CEMT SET DUMPDS 451  
 AUXILIARY  
   CEBT INQUIRE AUXTRACE 80  
   CEMT INQUIRE AUXTRACE 160  
   CEMT INQUIRE TSMODEL 348  
   CEMT INQUIRE TSQUEUE 354  
 auxiliary trace 533  
   CEBT 80  
   CEBT SET 91  
   CEMT 159  
   CEMT SET 414  
 AUXTRACE  
   CEBT 80  
   CEBT SET transaction 91  
   CEMT SET transaction 414  
   CEMT transaction 159  
 AVAILABLE  
   CEMT INQUIRE CONNECTION 176  
   CEMT INQUIRE DSNAME 218  
   CEMT INQUIRE MODENAME 264  
   CEMT SET DSNAME 447  
   CEMT SET MODENAME 468

## B

BACKLOG 322  
   CEMT SET TCPIPService 498  
 BACKOUT  
   CEMT INQUIRE CONNECTION 180  
   CEMT INQUIRE DSNAME 221  
   CEMT INQUIRE TASK 314  
   CEMT INQUIRE TRANSACTION 340  
   CEMT INQUIRE UOW 359  
   CEMT SET CONNECTION 419  
   CEMT SET DSNAME 447

BACKOUT (*continued*)  
     CEMT SET UOW 511  
 BACKUPNONBWO  
     CEMT INQUIRE UOWDSNFAIL 363  
 BACKUPTYPE  
     CEMT INQUIRE DSNAME 218  
 BASE  
     CEMT INQUIRE DSNAME 220  
 BASEDSNAME  
     CEMT INQUIRE DSNAME 219  
 basic mapping support (BMS)  
     logical message 612  
     page retrieval 609  
     paging commands 609  
 BDAM  
     CEMT INQUIRE DSNAME 216  
     CEMT INQUIRE FILE 237  
 BEAN  
     CEMT PERFORM STATISTICS 408  
 BEAN command  
     CEMT transaction 161  
 BEAN terminal definitions  
     CEMT INQUIRE BEAN 161  
 BEANNAME  
     CEMT INQUIRE REQUESTMODEL 292  
 BEFORE  
     CEDA 110  
 BELOW  
     CEMT INQUIRE PROGRAM 288  
     CEMT INQUIRE TRANSACTION 343  
 BINDING  
     CEMT INQUIRE WEBSERVICE 385  
 blank fields in a display 141  
 BMS (basic mapping support) 609  
 BMS paging  
     copy command 616  
     display routed pages 617  
     logical message 615  
     status of hardcopy terminal 616  
 BREXIT  
     CEMT INQUIRE TRANSACTION 340  
 BRFACILITY  
     CEMT INQUIRE TASK 313  
     CEMT SET transaction 416  
 BRIDGE  
     CEMT INQUIRE TASK 313  
 bridge facilities  
     CEMT INQUIRE requests 163  
 Bridge facilities  
     CEMT SET requests 416  
 BROWSE  
     CEMT INQUIRE FILE 237  
     CEMT SET FILE 455  
 BROWSESTATUS  
     CEMT INQUIRE FILE 237  
 BSAM terminals 567  
 BTS browser transaction 57

## C

C  
 CEMT INQUIRE PROGRAM 289  
 CACHE  
     CEMT INQUIRE UOWDSNFAIL 362  
 CACHEFREE  
     CEMT INQUIRE CLASSCACHE 170  
 CACHESIZE  
     CEMT INQUIRE CLASSCACHE 170  
     CEMT PERFORM CLASSCACHE command 394  
 CADP transaction 37, 623  
 CANCEL  
     CEMT INQUIRE CONNECTION 178  
     CEMT SET CONNECTION 419  
     CEMT SET TERMINAL 503  
     CMSG 568  
 CANCELED - TEMP STRG ERR N  
     CMSG 580  
 canceling a message 564  
 canceling a transaction 2, 565, 567, 571, 619  
 CAQP transaction 623  
 CATA transaction 623  
 CATD transaction 623  
 CATP transaction 623  
 CATR transaction 623  
 CATS transaction 623  
 CAUSE  
     CEMT INQUIRE UOWDSNFAIL 362  
 CAUT transaction 623  
 CAVM (CICS availability manager) 88  
 CBAM transaction 57, 623  
 CBRA transaction 623  
 CCIN transaction 623  
 CCMF transaction 623  
 CCMS transaction 623  
 CCRL transaction 59, 623  
 CCU2 transaction 623  
 CDATAKEY  
     CEMT INQUIRE TRANSACTION 342  
 CDBC transaction 61, 623  
 CDBD transaction 623  
 CDBF transaction 623  
 CDBI transaction 65, 623  
 CDBM transaction 67, 623  
 CDBN transaction 623  
 CDBO transaction 623  
 CDBQ transaction 623  
 CDBT transaction 623  
 CDFS transaction 623  
 CDSASIZE  
     CEMT INQUIRE DSAS 211  
 CDST transaction 623  
 CDTS transaction 623  
 CEBR transaction 75, 623  
 CEBT transaction 623  
     AUXTRACE 80, 91  
     CICS availability manager (CAVM) 88  
     DUMP 82, 93  
     GTFTRACE 83, 94  
     INQISET 79  
     INTTRACE 84, 95

CEBT transaction *(continued)*  
 PERFORM 87  
 request formats 79  
 SHUTDOWN 88  
 SNAP 89  
 SURVEILLANCE 85, 96  
 TAKEOVER 86, 90, 97  
 XRF complex 79  
 CECI transaction 99, 623  
 CECS transaction 101, 623  
 CEDA transaction 103, 624  
 security checking 114  
 CEDB transaction 103, 624  
 CEDC transaction 103, 624  
 CEDF  
 CEMT INQUIRE PROGRAM 287  
 CEMT SET PROGRAM 479  
 CEDF transaction 121, 624  
 CEDFSTATUS  
 CEMT INQUIRE PROGRAM 287  
 CEDX transaction 624  
 CEGN transaction 624  
 CEHP transaction 624  
 CEHS transaction 624  
 CEID transaction 624  
 CEJR transaction 624  
 CEKL transaction 624  
 request formats 127  
 TASK 128, 132  
 CEMS transaction 624  
 CEMT INQUIRE CONNECTION 180  
 CEMT PERFORM CLASSCACHE command 392  
 CEMT PERFORM PIPELINE command 402  
 CEMT transaction 624  
 - character 135  
 ? character 135  
 \* symbol 139  
 + in CEMT syntax 139  
 + in scrolling 141  
 abbreviation of keywords 139  
 AUTINSTMODEL 154  
 AUTOINSTALL 156, 412  
 AUXTRACE 159, 414  
 BEAN 161  
 blank fields in a display 141  
 BRFACILITY 163, 416  
 CEMT as a prompter 136  
 CEMT on the command line 135  
 CFDTPOOL 167  
 CLASSCACHE 169, 417  
 CONNECTION 173, 418  
 CORBASERVER 182, 424  
 DB2CONN 186, 426  
 DB2ENTRY 434  
 DB2TRAN 198, 439  
 DBENTRY 193  
 DELETSHIPED 200, 440  
 DISCARD 145, 147  
 DISPATCHER 202, 442  
 DJAR 205  
 DOCTEMPLATE 208

CEMT transaction *(continued)*  
 DSAS 211, 445  
 DSNAME 214, 446  
 DUMP 400  
 DUMPDS 223, 451  
 ENDAFFINITY 401  
 ENQ 225  
 ENQMODEL 230  
 ENQUEUE 452  
 EXCI 232  
 Family of resources 139  
 FILE 235, 454  
 generic names 139  
 generic resource identifiers 139  
 GTFTRACE 242, 459  
 HOST 244, 460  
 INQUIRE 135  
 INTTRACE 246, 461  
 invoking from an application program 144  
 IRBATCH 248  
 IRC 249, 462  
 JMODEL 251  
 JOURNALNAME 253, 463  
 JVM 256  
 JVMPOOL 259, 464  
 LINE 261, 466  
 minus sign following CEMT 135  
 MODENAME 263, 468  
 MONITOR 265, 470  
 NETNAME 269, 473  
 overtyping a display 140  
 PARTNER 275  
 PERFORM 391  
 PIPELINE 277, 476  
 PROCESSTYPE 280, 477  
 PROFILE 282  
 PROGRAM 283, 479  
 program function (PF) keys 138  
 question mark preceding CEMT 135  
 QUEUE 482  
 request formats 135  
 REQUESTMODEL 291  
 RESET 403  
 RRMS 294  
 scrolling a display 141  
 SECURITY 404  
 security checking 144  
 SET 135  
 SHUTDOWN 405  
 SNAP 400  
 STATISTICS 295, 407, 483  
 STREAMNAME 299  
 SYDUMPCODE 301, 486  
 SYSTEM 305, 489  
 tab key 141  
 TASK 311, 493  
 TCLASS 317, 495  
 TCPIP 319, 496  
 TCPIPSERVICE 320, 498  
 TDQUEUE 326, 500  
 TERMINAL 331, 502

CEMT transaction (*continued*)

- TRANSACTION 338, 506
- TRDUMPCODE 344, 508
- TSMODEL 347
- TSPPOOL 350
- TSQNAME 352
- TSQUEUE 352, 510
- UOW 356, 511
- UOWDSNFAIL 361
- UOWENQ 366
- UOWLINK 371, 512
- URIMAP 375, 513
- VTAM 380, 515
- WEB 382, 517
- WEBSERVICE 383, 518
- without keywords 138
- WORKREQUEST 387, 519

CEOS transaction 624

CEOT transaction 521, 624

CEPC transaction 624

CEPE transaction 624

CEPQ transaction 624

CEPW transaction 624

CERTIFICATE 322

- CEMT INQUIRE CORBASERVER 183
- CEMT INQUIRE URIMAP 378

certificate revocation list transaction (CCRL) 59

CESC transaction 624

CESD transaction 624

CESF transaction 525, 624

CESN 529, 530

CESN transaction 527, 624

CEST transaction 531, 624

CETR transaction 533, 624

CEX2 transaction 624

CEXECKEY

- CEMT INQUIRE PROGRAM 288

CFCL transaction 624

CFDTPPOOL

- CEMT INQUIRE CFDTPPOOL 167
- CEMT SET FILE 455
- CEMT transaction 167

CFOR transaction 624

CFQS transaction 624

CFTABLE

- CEMT INQUIRE FILE 241
- CEMT INQUIRE UOWLINK 373
- CEMT SET FILE 455

CFTI transaction 624

CFTL transaction 624

CFTM transaction 624

CFTS transaction 624

CFUP transaction 624

CGROUP

- CEMT INQUIRE DB2CONN 188
- CEMT SET DB2CONN 428

CGRP transaction 624

chaining a message

- CSPG transaction 612
- partitions 8

changing debugging profiles

- with the 3270 interface 40
- with the web interface 22

CHARACTERSET

- CEMT INQUIRE URIMAP 378

checking user storage 607

CHKSTRM

- CSFE DEBUG 608

CHLP transaction 624

CICS availability manager (CAVM) 88

CICS DB2 transaction (DSNC) 621

CICS releases, CSD compatibility across 112

CICSAPI

- CEMT INQUIRE PROGRAM 286

CICSTABLE

- CEMT INQUIRE FILE 241
- CEMT SET FILE 455

CICSTELEVEL

- CEMT INQUIRE SYSTEM 306

CID

- CEMT INQUIRE NETNAME 271
- CEMT INQUIRE TERMINAL 334

CIDP transaction 551, 624

CIEP transaction 624

CIND transaction 553, 624

CINS transaction 624

CIOD transaction 624

CIOF transaction 624

CIOR transaction 624

CIPHERS

- CEMT INQUIRE URIMAP 378

CIRB transaction 624

CIRD transaction 624

CIRP transaction 624

CIRR transaction 625

CITS transaction 625

CJMJ transaction 625

CJTR transaction 625

CKOPEN

- CEMT INQUIRE TASK 316

CLASS

- CEMT INQUIRE PROGRAM 283
- CEMT INQUIRE TERMINAL 332
- CEMT INQUIRE TRANSACTION 338
- CEMT SET PROGRAM 480
- CEMT SET TERMINAL 503
- CEMT SET TRANSACTION 506
- general statement 139

class tasks, CEMT requests 317, 495

CLASSCACHE

- CEMT SET CLASSCACHE 417
- CEMT transaction 169, 392

CLASSCACHST

- CEMT INQUIRE JVM 257

CLEAR key

- when invoking transaction 2
- when used with partitions 8

CLEAR PARTITION key 8

CLEAR/CANCL or CONT INPUT/ROUTE/MSG

- CMSG 581

CLEP transaction 625

CLER transaction 557, 625  
 CLIENT  
   CEMT INQUIRE URIMAP 377  
 CLIENTCERT  
   CEMT INQUIRE CORBASERVER 184  
 CLIENTIPADDR option  
   INQUIRE WORKREQUEST command 389  
 CLOSED  
   CEBT INQUIRE DUMP 82  
   CEBT SET DUMP 93  
   CEMT INQUIRE DUMPDS 224  
   CEMT INQUIRE FILE 240  
   CEMT INQUIRE IRC 250  
   CEMT INQUIRE MODENAME 264  
   CEMT INQUIRE RRMS 294  
   CEMT INQUIRE TDQUEUE 329  
   CEMT INQUIRE VTAM 381  
   CEMT SET DUMPDS 451  
   CEMT SET FILE 455  
   CEMT SET IRC 462  
   CEMT SET MODENAME 469  
   CEMT SET TDQUEUE 501  
   CEMT SET VTAM 515  
 CLOSETIMEOUT  
   INQUIRE TCPIPService 322  
 CLQ2 transaction 625  
 CLR1 transaction 625  
 CLR2 transaction 625  
 CLS1 transaction 625  
 CLS2 transaction 625  
 CLS3 transaction 625  
 CLS4 transaction 625  
 CMAC transaction 559, 625  
 CMDPROT  
   CEMT INQUIRE SYSTEM 306  
 CMDPROTECT  
   CEMT INQUIRE SYSTEM 306  
 CMPX transaction 625  
 CMSG transaction 563, 625  
   CANCEL 568  
   DATE 568  
   ERRTERM 570  
   FULLDATE 570  
   HEADING 571  
   ID 571  
   logical unit destination 576  
   MSG 571  
   OPCLASS 572  
   options 568  
   program (DFHMSP) 580  
   PROTECT 572  
   ROUTE 572  
   SEND 575  
   syntax 568  
   TIME 575  
 CMTS transaction 625  
 COBOL  
   CEMT INQUIRE PROGRAM 289  
 codes, display 559  
 COLD  
   CEMT INQUIRE UOWLINK 373  
 COLDACQ  
   CEMT INQUIRE NETNAME 274  
   CEMT INQUIRE TERMINAL 337  
   CEMT SET NETNAME 474  
   CEMT SET TERMINAL 503  
 COMAUTHID  
   CEMT INQUIRE DB2CONN 188  
   CEMT SET DB2CONN 428  
 COMAUTHTYPE  
   CEMT INQUIRE DB2CONN 188  
 COMMAND  
   CEBT SET TAKEOVER 97  
 command interpreter transaction (CECI) 99  
 command syntax checker (CECS) 101  
 commands  
   CEBT INQISET 79  
   CEBT PERFORM 87  
   CEMT DISCARD 147  
   CEMT INQUIRE 152  
   CEMT PERFORM 391  
   CEMT SET 411  
   paging 609  
 COMMIT  
   CEMT INQUIRE CONNECTION 180  
   CEMT INQUIRE DSNAME 221  
   CEMT INQUIRE TASK 314  
   CEMT INQUIRE TRANSACTION 340  
   CEMT INQUIRE UOW 359  
   CEMT SET CONNECTION 419  
   CEMT SET DSNAME 447  
   CEMT SET UOW 511  
 COMMITFAIL  
   CEMT INQUIRE UOWDSNFAIL 363  
 communicating with CICS 1  
 compatibility, CSD, across CICS releases 112  
 COMPLETE  
   CEMT INQUIRE CONNECTION 177  
 component trace options 533  
 COMTHREADLIM  
   CEMT INQUIRE DB2CONN 189  
   CEMT SET DB2CONN 429  
 COMTHREADS  
   CEMT INQUIRE DB2CONN 189  
 CONCURRENCY  
   CEMT INQUIRE PROGRAM 287  
 CONFIGFILE  
   CEMT INQUIRE PIPELINE 278  
 CONNECT  
   CEMT INQUIRE DB2CONN 191  
   CEMT SET DB2CONN 432  
 CONNECTED  
   CEMT INQUIRE DB2CONN 189  
   CEMT INQUIRE TSPool 351  
   CEMT SET DB2CONN 429  
 CONNECTERROR  
   CEMT INQUIRE DB2CONN 189  
   CEMT SET DB2CONN 429  
 CONNECTION  
   CEDA 110  
   CEMT DISCARD 147  
   CEMT INQUIRE CONNECTION 176

CONNECTION *(continued)*  
   CEMT INQUIRE MODENAME 264  
   CEMT INQUIRE UOW 360  
   CEMT INQUIRE UOWDSNFAIL 362  
   CEMT INQUIRE UOWLINK 373  
   CEMT PERFORM STATISTICS 409  
   CEMT SET CONNECTION 418  
   CEMT SET MODENAME 469  
 CONNECTST  
   CEMT INQUIRE DB2CONN 189  
   CEMT SET DB2CONN 429  
 CONNSTATUS  
   CEMT INQUIRE CFDTPOOL 167  
   CEMT INQUIRE CONNECTION 176  
   CEMT INQUIRE TSPOOL 351  
 CONSOLE  
   CEMT INQUIRE TERMINAL 334  
 console as a CICS terminal 9  
 console operator, write to 619  
 console support, multiple 9  
 CONSOLES  
   CEMT INQUIRE AUTOINSTALL 157  
   CEMT SET AUTOINSTALL 412  
 CONTAINER  
   CEMT INQUIRE WEBSERVICE 386  
 CONTENTION  
   CEMT SET FILE 455  
 CONTINUE INPUT  
   MSG 581  
 CONTINUE INPUT OR MSG  
   MSG 581  
 CONTINUE MSG  
   MSG 581  
 CONTINUE ROUTE OPERAND  
   MSG 581  
 CONVERSE  
   CEMT INQUIRE MONITOR 266  
   CEMT SET MONITOR 471  
 CONVERSEST  
   CEMT INQUIRE MONITOR 266  
 CONVERTER  
   CEMT INQUIRE URIMAP 377  
 COORDINATOR  
   CEMT INQUIRE UOWLINK 373  
 COPID  
   CEMT INQUIRE DB2CONN 188  
   CEMT SET DB2CONN 429  
   option of DSNCRCT macro 429  
 COPY command 616  
 copying debugging profiles  
   with the 3270 interface 41  
   with the web interface 23  
 copying pages  
   CSPG 616  
   partitions 9  
 COPYSTATUS  
   CEMT INQUIRE PROGRAM 287  
 CORBASERVER  
   CEDA 110  
   CEMT DISCARD 148  
   CEMT INQUIRE BEAN 162  
 CORBASERVER *(continued)*  
   CEMT INQUIRE DJAR 206  
   CEMT INQUIRE REQUESTMODEL 292  
   CEMT PERFORM CORBASERVER 396  
   CEMT PERFORM STATISTICS 409  
   CEMT SET CORBASERVER 424  
   CEMT transaction 182, 396  
 CORBASERVER option  
   INQUIRE WORKREQUEST command 389  
 COVR transaction 625  
 CPIA transaction 589  
 CPIS transaction 625  
 CPLT transaction 625  
 CPMI transaction 625  
 CQP  
   CEMT INQUIRE CONNECTION 176  
 CQPI transaction 625  
 CQPO transaction 625  
 CQRY transaction 625  
 CRDR transaction 625  
 CRE  
   CEOT 522  
 CREA transaction 591, 625  
 CREATE  
   CEMT INQUIRE NETNAME 271  
   CEMT INQUIRE TERMINAL 334  
   CEMT SET NETNAME 474  
   CEMT SET TERMINAL 504  
 Create Compiled Debugging Profile screen  
   application debugging profile manager  
     3270 interface 42  
 Create compiled profile page  
   application debugging profile manager  
     web interface 24  
 Create Java Debugging Profile screen  
   application debugging profile manager  
     3270 interface 47  
 Create Java profile page  
   application debugging profile manager  
     web interface 28  
 CREATSESS  
   CEMT INQUIRE NETNAME 271  
   CEMT INQUIRE TERMINAL 334  
 creating debugging profiles  
   with the 3270 interface 39  
   with the web interface 21  
 CREC transaction 591, 625  
 CRITICAL value 323  
 CRLPROFILE  
   CEMT INQUIRE TCPIP 319  
 CRMD transaction 625  
 CRMF transaction 625  
 CRPA transaction 625  
 CRPC transaction 625  
 CRPM transaction 625  
 CRSQ transaction 625  
 CRSR transaction 625  
 CRSY transaction 625  
 CRTE transaction 603, 625  
 CRTX transaction 625  
 CSAC transaction 625

CSCY transaction 625  
 CSD compatibility across CICS releases 112  
 CSFE transaction 605, 625  
 CSFR transaction 625  
 CSFU transaction 625  
 CSGM transaction 625  
 CSHA transaction 625  
 CSHQ transaction 626  
 CSHR transaction 626  
 CSIGN  
     CEMT INQUIRE DB2CONN 188  
     CEMT SET DB2CONN 428  
 CSIR 160 transaction 626  
 CSJC transaction 626  
 CSKD transaction 626  
 CSKE transaction 626  
 CSKL transaction 626  
 CSKP transaction 626  
 CSM1 transaction 626  
 CSM2 transaction 626  
 CSM3 transaction 626  
 CSM5 transaction 626  
 CSMI transaction 626  
 CSMT transaction 626  
 CSNC transaction 626  
 CSNE transaction 626  
 CSOL transaction 626  
 CSOT transaction 626  
 CSPG transaction 609, 626  
 CSPK transaction 626  
 CSPP transaction 626  
 CSPQ transaction 626  
 CSPS transaction 626  
 CSQC transaction 626  
 CSRK transaction 626  
 CSRS transaction 626  
 CSSC transaction 626  
 CSSF transaction 626  
 CSSN transaction 626  
 CSST transaction 626  
 CSSY transaction 626  
 CSTA transaction 626  
 CSTB transaction 626  
 CSTE transaction 626  
 CSTP transaction 626  
 CSTT transaction 626  
 CSZI transaction 626  
 CTERM  
     CEMT INQUIRE DB2CONN 188  
     CEMT SET DB2CONN 428  
     DSNCRCT macro 428  
 CTIN transaction 626  
 CTLGALL  
     CEMT INQUIRE SYSTEM 308  
     CEMT SET SYSTEM 490  
 CTLGMODIFY  
     CEMT INQUIRE SYSTEM 308  
     CEMT SET SYSTEM 490  
 CTLGNONE  
     CEMT INQUIRE SYSTEM 308  
     CEMT SET SYSTEM 490

CTSD transaction 626  
 CTX  
     CEMT INQUIRE DB2CONN 188  
     CEMT SET DB2CONN 429  
 CURAUXDS  
     CEMT INQUIRE AUXTRACE 160  
 CURRENT  
     CEMT INQUIRE SYDUMPCODE 303  
     CEMT INQUIRE TRDUMPCODE 345  
     CSFE 607  
 CURRENT(AIB)  
     CEBT INQUIRE AUXTRACE 80  
 CURRENTDDS  
     CEBT INQUIRE DUMP 82  
     CEMT INQUIRE DUMPDS 224  
 CURREQS  
     CEMT INQUIRE AUTOINSTALL 157  
 CUSERID  
     CEMT INQUIRE DB2CONN 188  
     CEMT SET DB2CONN 429  
 CVMI transaction 626  
 CVST transaction 626  
 CWBA transaction 626  
 CWBC transaction 626  
 CWBG transaction 626  
 CWBM transaction 626  
 CWTO transaction 619, 626  
 CWTR transaction 626  
 CWXN transaction 626  
 CWXU transaction 627  
 CXCU transaction 627  
 CXRE transaction 627  
 CXRT transaction 627

## D

D  
     CEMT INQUIRE TASK 315  
 DAE  
     CEMT INQUIRE SYDUMPCODE 303  
     CEMT SET SYDUMPCODE 487  
 DAEOPTION  
     CEMT INQUIRE SYDUMPCODE 303  
 data sets  
     CEMT requests 235  
     CEMT SET requests 454  
 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 327  
 DATALOC  
     CEMT INQUIRE PROGRAM 288  
 DATASET  
     CEMT INQUIRE ENQ 228  
     CEMT INQUIRE UOW 360  
     CEMT INQUIRE UOWDSNFAIL 363  
     CEMT INQUIRE UOWENQ 369

DATASETFULL  
     CEMT INQUIRE UOWDSNFAIL 363  
 DATE  
     CMSG 568  
 DATE ALREADY PASSED  
     CMSG 581  
 DATE INVALID AT XXXXXXXX  
     CMSG 581  
 DATE NOT XX/XX AT XXXXX  
     CMSG 582  
 date resetting 403  
 DATE TOO FAR IN FUTURE  
     CMSG 582  
 DATESTAMP  
     CEMT INQUIRE DJAR 206  
 DATESTARTED  
     CEMT INQUIRE CLASSCACHE 170  
 DATFORM, system initialization parameter 7  
 DB2  
     CEMT PERFORM STATISTICS 409  
 DB2 definitions  
     connection  
         CEMT INQUIRE DB2CONN 186  
         CEMT SET DB2CONN command 426  
     entry  
         CEMT INQUIRE DB2ENTRY 193  
         CEMT SET DB2ENTRY command 434  
     transaction  
         CEMT INQUIRE DB2TRAN 198  
         CEMT SET DB2TRAN command 439  
 DB2CONN  
     CEDA 110  
     CEMT DISCARD 148  
     CEMT INQUIRE DB2CONN 186  
     CEMT INQUIRE SYSTEM 306  
     CEMT SET transaction 426  
     CEMT transaction 186  
 DB2ENTRY  
     CEDA 110  
     CEMT DISCARD 148  
     CEMT INQUIRE DB2ENTRY 193  
     CEMT INQUIRE DB2TRAN 198  
     CEMT SET DB2TRAN 439  
     CEMT SET transaction 434  
     CEMT transaction 193  
 DB2GROUPID  
     CEMT INQUIRE DB2CONN 189  
     CEMT SET DB2CONN 430  
 DB2ID  
     CEMT INQUIRE DB2CONN 189  
     CEMT SET DB2CONN 430  
 DB2PLAN  
     CEMT INQUIRE TASK 314  
 DB2RELEASE  
     CEMT INQUIRE DB2CONN 189  
 DB2TRAN  
     CEDA 110  
     CEMT DISCARD 148  
     CEMT INQUIRE DB2TRAN 198  
     CEMT SET transaction 439  
     CEMT transaction 198  
 DBCTL (database control) 61  
 DBCTL system identifier (DBCTLID) 63  
 DBCTLID 63  
 DDNAME  
     CEMT INQUIRE DOCTEMPLATE 209  
     CEMT INQUIRE TDQUEUE 328  
 DEADLOCK  
     CEMT INQUIRE UOWDSNFAIL 363  
 DEBUG  
     CEMT SET SYSTEM 490  
     CSFE 606  
 debugging profile manager  
     3270 interface 37  
         activating debugging profiles 40  
         changing debugging profiles 40  
         copying debugging profiles 41  
         Create Compiled Debugging Profile screen 42  
         Create Java Debugging Profile screen 47  
         creating debugging profiles 39  
         deleting debugging profiles 41  
         inactivating debugging profiles 40  
         List Debugging Profiles screen 37  
         Set Compiled Debugging Display Device  
             screen 54  
         setting the display device 53  
         View Compiled Debugging Profile screen 50  
         View Java Debugging Profile screen 51  
         viewing debugging profiles 40  
     web interface 16  
         activating debugging profiles 21  
         changing debugging profiles 22  
         configuring 16  
         copying debugging profiles 23  
         Create compiled profile page 24  
         Create Java profile page 28  
         creating debugging profiles 21  
         deleting debugging profiles 23  
         inactivating debugging profiles 22  
         List all profiles page 17  
         Set compiled display device page 34  
         setting the display device 33  
         View compiled profile page 31  
         View Java profile page 32  
         viewing debugging profiles 22  
 debugging profile manager, application 15  
 debugging profiles  
     activating  
         with the 3270 interface 40  
         with the web interface 21  
     changing  
         with the 3270 interface 40  
         with the web interface 22  
     copying  
         with the 3270 interface 41  
         with the web interface 23  
     creating  
         with the 3270 interface 39  
         with the web interface 21  
     deleting  
         with the 3270 interface 41  
         with the web interface 23

debugging profiles *(continued)*  
 inactivating  
   with the 3270 interface 40  
   with the web interface 22  
 viewing  
   with the 3270 interface 40  
   with the web interface 22

DEBUGTOOL  
 CEMT INQUIRE SYSTEM 306

DEL  
 CEMT INQUIRE DELETSHIPED 201

DELETE  
 CEMT INQUIRE FILE 238  
 CEMT SET FILE 455  
 CEMT SET TSQUEUE 510  
 CEMT SET UOWLINK 512

DELETESTATUS  
 CEMT INQUIRE FILE 238

deleting debugging profiles  
 with the 3270 interface 41  
 with the web interface 23

DELETSHIPED  
 CEMT SET transaction 440  
 CEMT transaction 398

DELETSHIPED command  
 CEMT transaction 200

DELEXITERROR  
 CEMT INQUIRE UOWDSNFAIL 363

DEREGERROR  
 INQUIRE TCPIPService 323

DEREGISTERED  
 CEMT INQUIRE VTAM 381  
 CEMT SET VTAM 515  
 INQUIRE TCPIPService 323

DEST  
 CEMT INQUIRE TASK 314

destinations, multiple 563

DFHCICSP 4

DFLTUSER  
 CEMT INQUIRE SYSTEM 307

DISABLED  
 CEMT INQUIRE CORBASERVER 184  
 CEMT INQUIRE DB2ENTRY 196  
 CEMT INQUIRE ENQMODEL 231  
 CEMT INQUIRE FILE 239  
 CEMT INQUIRE HOST 245  
 CEMT INQUIRE JOURNALNAME 254  
 CEMT INQUIRE PROGRAM 290  
 CEMT INQUIRE TDQUEUE 328  
 CEMT INQUIRE TRANSACTION 342  
 CEMT INQUIRE URIMAP 377  
 CEMT PERFORM CLASSCACHE command 394  
 CEMT SET CLASSCACHE command 417  
 CEMT SET CORBASERVER 424  
 CEMT SET DB2ENTRY 437  
 CEMT SET ENQMODEL 452  
 CEMT SET FILE 455  
 CEMT SET HOST 460  
 CEMT SET JOURNALNAME 463  
 CEMT SET JVMPOOL 464  
 CEMT SET PROCESSTYPE 478

DISABLED *(continued)*  
 CEMT SET PROGRAM 480  
 CEMT SET TDQUEUE 501  
 CEMT SET TRANSACTION 506  
 CEMT SET URIMAP 513

DISABLEDACT  
 CEMT INQUIRE DB2ENTRY 195  
 CEMT SET DB2ENTRY 437

DISABLING  
 CEMT INQUIRE CORBASERVER 184  
 CEMT INQUIRE DB2ENTRY 196

DISCARDING  
 CEMT INQUIRE CORBASERVER 184

DISPATCHABLE  
 CEKL INQUIRE TASK 129  
 CEMT INQUIRE TASK 315

DISPATCHER  
 CEMT INQUIRE transaction 202  
 CEMT PERFORM STATISTICS 409  
 CEMT SET transaction 442

display device  
 setting  
   with the 3270 interface 53  
   with the web interface 33

display logical message (BMS) 612

display routed pages 617

display transaction 559

DISPOSITION  
 CEMT INQUIRE FILE 238  
 CEMT INQUIRE TDQUEUE 328

DJAR  
 CEDA 110  
 CEMT DISCARD 148  
 CEMT INQUIRE BEAN 162  
 CEMT PERFORM DJAR 399  
 CEMT transaction 205, 399  
 CREA 592

DJARDIR  
 CEMT INQUIRE CORBASERVER 184

DNSGROUP  
 INQUIRE TCPIPService 322

DNSSTATUS  
 CEMT SET TCPIPService 498

DNSSTATUS value  
 INQUIRE TCPIPService 322

DOCTEMPLATE  
 CEDA 110  
 CEMT DISCARD 149  
 CEMT INQUIRE DOCTEMPLATE 209  
 CEMT transaction 208

DPLSUBSET  
 CEMT INQUIRE PROGRAM 289  
 CEMT SET PROGRAM 480

DROLLBACK  
 CEMT INQUIRE DB2CONN 189  
 CEMT INQUIRE DB2ENTRY 196

DS  
 CEMT INQUIRE TASK 315

DSALIMIT  
 CEMT INQUIRE DSAS 211  
 CEMT INQUIRE SYSTEM 307

DSALIMIT (*continued*)  
   CEMT SET DSAS 445  
   CEMT SET SYSTEM 490  
 DSAS  
   CEMT INQUIRE transaction 211  
   CEMT SET transaction 445  
 DSNAME  
   CEMT INQUIRE DOCTEMPLATE 209  
   CEMT INQUIRE DSNAME 219  
   CEMT INQUIRE FILE 238  
   CEMT INQUIRE TDQUEUE 328  
   CEMT INQUIRE UOWDSNFAIL 363  
   CEMT SET FILE 455  
   CEMT SET transaction 446  
   CEMT transaction 214  
 DSNL transaction 621  
 DSNL transactions  
   DISCONNECT 621  
   DISPLAY 621  
   MODIFY 621  
   STOP 621  
   STRT 621  
 DSRTPROGRAM  
   CEMT SET SYSTEM 491  
 DTRPROGRAM  
   CEMT INQUIRE SYSTEM 307  
   CEMT SET SYSTEM 491  
 DUMMY  
   CEMT INQUIRE JMODEL 252  
   CEMT INQUIRE JOURNALNAME 255  
 DUMP  
   CEBT 82  
   CEBT INQUIRE DUMP 82  
   CEBT PERFORM SHUTDOWN 88  
   CEBT PERFORM TAKEOVER 90  
   CEBT SET transaction 93  
   CEMT PERFORM DUMP 400  
   CEMT PERFORM SHUTDOWN 405  
   CEMT transaction 400  
 DUMPDS  
   CEMT INQUIRE DUMPDS 224  
   CEMT SET transaction 451  
   CEMT transaction 223  
 DUMPING  
   CEMT INQUIRE SYSTEM 307  
 DUMPSCOPE  
   CEMT INQUIRE SYDUMPCODE 303  
   CEMT INQUIRE TRDUMPCODE 345  
 DURATION  
   CEMT INQUIRE ENQ 227  
   CEMT INQUIRE UOWENQ 368  
 DYNAMIC  
   CEMT INQUIRE DSNAME 218  
   CEMT INQUIRE PROGRAM 288  
 DYNAMSTATUS  
   CEMT INQUIRE PROGRAM 288  
  
**E**  
 ECDSASIZE  
   CEMT INQUIRE DSAS 212  
  
 EDF (execution diagnostic facility) 121, 603  
 EDSALIMIT  
   CEMT INQUIRE DSAS 212  
   CEMT INQUIRE SYSTEM 307  
   CEMT SET DSAS 445  
   CEMT SET SYSTEM 491  
 EMPTYREQ  
   CEMT INQUIRE FILE 238  
   CEMT SET FILE 456  
 EMPTYSTATUS  
   CEMT INQUIRE FILE 238  
 ENABLED  
   CEMT INQUIRE CORBASERVER 184  
   CEMT INQUIRE DB2ENTRY 196  
   CEMT INQUIRE ENQMODEL 231  
   CEMT INQUIRE FILE 238  
   CEMT INQUIRE HOST 245  
   CEMT INQUIRE JOURNALNAME 254  
   CEMT INQUIRE PROGRAM 290  
   CEMT INQUIRE TDQUEUE 328  
   CEMT INQUIRE TRANSACTION 342  
   CEMT INQUIRE URIMAP 377  
   CEMT PERFORM CLASSCACHE command 394  
   CEMT SET CLASSCACHE command 417  
   CEMT SET CORBASERVER 425  
   CEMT SET DB2ENTRY 437  
   CEMT SET ENQMODEL 452  
   CEMT SET FILE 456  
   CEMT SET HOST 460  
   CEMT SET JOURNALNAME 463  
   CEMT SET JVMPOOL 464  
   CEMT SET PROCESSTYPE 478  
   CEMT SET PROGRAM 480  
   CEMT SET TDQUEUE 501  
   CEMT SET TRANSACTION 507  
   CEMT SET URIMAP 513  
 ENABLESTATUS  
   CEMT INQUIRE AUTOINSTALL 157  
   CEMT INQUIRE CORBASERVER 184  
   CEMT INQUIRE DB2ENTRY 196  
   CEMT INQUIRE FILE 238  
   CEMT INQUIRE HOST 245  
   CEMT INQUIRE PIPELINE 278, 476  
   CEMT INQUIRE PROCESSTYPE 281  
   CEMT INQUIRE TDQUEUE 328  
   CEMT INQUIRE URIMAP 377  
   CEMT SET DB2ENTRY 437  
 ENABLING  
   CEMT INQUIRE CORBASERVER 184  
 END  
   CSFE 605  
 ENDAFFINITY  
   CEMT INQUIRE CONNECTION 176  
   CEMT SET CONNECTION 419  
 ENDOFDAY  
   CEMT INQUIRE STATISTICS 296  
   CEMT SET STATISTICS 483  
 ENDPOINT  
   CEMT INQUIRE WEBSERVICE 385  
 ENQ  
   CEMT transaction 225

ENQFAILS  
 CEMT INQUIRE ENQ 227  
 CEMT INQUIRE UOWENQ 368

ENQMODEL  
 CEDA 110  
 CEMT DISCARD 149  
 CEMT INQUIRE ENQMODEL 231  
 CEMT transaction 230

ENQNAME  
 CEMT INQUIRE ENQMODEL 231

ENQQUEUE  
 CEMT SET transaction 452

ENQSCOPE  
 CEMT INQUIRE ENQ 227  
 CEMT INQUIRE ENQMODEL 231  
 CEMT INQUIRE UOWENQ 368

ENQUEUE  
 CEMT PERFORM STATISTICS 409

EQUAL  
 CEMT INQUIRE DB2CONN 190, 196  
 CEMT SET DB2CONN 431  
 CEMT SET DB2ENTRY 437

ERDSASIZE  
 CEMT INQUIRE DSAS 212  
 error message in partition 9  
 errors in message switching 579

ERRTERM  
 CMSG 570

ERRTERM INVALID AT XXXXX  
 CMSG 582

ERRTERM INVLD/UNSUPP XXXX  
 CMSG 582

ESDSASIZE  
 CEMT INQUIRE DSAS 212

EUDSASIZE  
 CEMT INQUIRE DSAS 212

EXCEPT  
 CEMT INQUIRE MONITOR 266  
 CEMT SET MONITOR 471

EXCEPTCLASS  
 CEMT INQUIRE MONITOR 266

EXCI  
 CEMT INQUIRE CONNECTION 178  
 CEMT INQUIRE EXCI 233  
 CEMT INQUIRE transaction 232

EXCLSTATUS  
 CEMT INQUIRE FILE 239

EXCLUSIVE  
 CEMT INQUIRE FILE 239  
 CEMT SET FILE 456

EXECENQ  
 CEMT INQUIRE ENQ 228  
 CEMT INQUIRE UOWENQ 369

EXECKEY  
 CEMT INQUIRE JVM 257  
 CEMT INQUIRE PROGRAM 288

execution diagnostic facility (EDF) 121

EXECUTIONSET  
 CEMT INQUIRE PROGRAM 289

EXTRA  
 CEMT INQUIRE TDQUEUE 330

**F**

FACILITY  
 CEMT INQUIRE TASK 314

FACILITYLIKE  
 CEMT INQUIRE TRANSACTION 340

FAILED  
 CEMT INQUIRE CONNECTION 177  
 CEMT INQUIRE JOURNALNAME 254  
 CEMT INQUIRE STREAMNAME 300

FAILEDDBKOUT  
 CEMT INQUIRE UOWDSNFAIL 363

Family of resources 139

FCANCEL  
 CEMT INQUIRE CONNECTION 179  
 CEMT SET CONNECTION 420

FENODE  
 CEMT DISCARD 149

FEPI  
 CEMT PERFORM STATISTICS 409

FEPI, Front End Programming Interface xi

FEPOOL  
 CEMT DISCARD 149

FEPROPSET  
 CEMT DISCARD 149

FETARGET  
 CEMT DISCARD 149

FILE  
 CEDA 110  
 CEMT DISCARD 149  
 CEMT INQUIRE ENQ 228  
 CEMT INQUIRE FILE 239  
 CEMT INQUIRE PROCESSTYPE 281  
 CEMT INQUIRE UOWENQ 369  
 CEMT PERFORM STATISTICS 409  
 CEMT SET transaction 454  
 CEMT transaction 235

FILECOUNT  
 CEMT INQUIRE DSNAME 219

FILELIMIT  
 CEMT INQUIRE MONITOR 266

fixed-format terminals 525

FLUSH  
 CEMT INQUIRE JOURNALNAME 254  
 CEMT SET JOURNALNAME 463

FORCE  
 CEMT INQUIRE SYSTEM 307  
 CEMT INQUIRE UOW 359  
 CEMT SET DSNAME 447  
 CEMT SET SYSTEM 491  
 CEMT SET UOW 511

FORCECLOSE  
 CEMT INQUIRE FILE 240  
 CEMT INQUIRE VTAM 381  
 CEMT SET FILE 456  
 CEMT SET VTAM 515

FORCEDISABLED  
 CEMT SET DB2ENTRY 437

FORCENOTCON  
 CEMT SET DB2CONN 430

FORCEPURGE  
 CEKL SET TASK 132

FORCEPURGE *(continued)*  
 CEMT INQUIRE CONNECTION 178  
 CEMT INQUIRE NETNAME 272  
 CEMT INQUIRE TASK 315  
 CEMT INQUIRE TERMINAL 336  
 CEMT PERFORM CLASSCACHE command 393  
 CEMT SET CONNECTION 420  
 CEMT SET JVMPOOL 464  
 CEMT SET LINE 466  
 CEMT SET NETNAME 474  
 CEMT SET TASK 493  
 CEMT SET TERMINAL 504  
 CEMT SET WORKREQUEST 519  
 INQUIRE WORKREQUEST command 389

FORCEQR  
 CEMT INQUIRE SYSTEM 307  
 CEMT SET SYSTEM 491, 492

FORCEUOW  
 CEMT INQUIRE CONNECTION 180  
 CEMT SET CONNECTION 420

FREEING  
 CEMT INQUIRE CONNECTION 176

FREQUENCY  
 CEMT INQUIRE MONITOR 266  
 CEMT SET MONITOR 471

FRLOG  
 CEMT INQUIRE DSNAME 219

Front End Programming Interface (FEPI) xi

FTYPE  
 CEMT INQUIRE TASK 314

FULL  
 CEMT SET PROCESSTYPE 477

FULLAPI  
 CEMT INQUIRE PROGRAM 289  
 CEMT SET PROGRAM 480

FULLDATE  
 CMSG 570

FWDRECOVABLE  
 CEMT INQUIRE DSNAME 221

FWDRECOVLSN  
 CEMT INQUIRE DSNAME 219

## G

GARBAGEINT  
 CEMT INQUIRE WEB 382  
 CEMT SET WEB 517

generic names 139  
 generic resource identifiers 139

GETPAGE  
 CSPG 609, 612

global kernel exit routines 608  
 global trap/trace program exit routine 608

GMT  
 CEMT INQUIRE MONITOR 267

GRNAME  
 CEMT INQUIRE CONNECTION 177  
 CEMT INQUIRE VTAM 381

GROUP  
 CEDA 110  
 CEMT INQUIRE DB2ENTRY 195

GROUP *(continued)*  
 CEMT SET DB2CONN 427  
 CEMT SET DB2ENTRY 436  
 group identifier 529  
 GROUPLD 529  
 GRPCRITICAL value  
 INQUIRE TCPIP SERVICE 323  
 GRSTATUS  
 CEMT INQUIRE VTAM 381  
 GTF trace 533  
 GTFTRACE  
 CEBT INQUIRE GTFTRACE 83  
 CEBT INQUIRE transaction 83  
 CEBT SET transaction 94  
 CEMT INQUIRE GTFTRACE 243  
 CEMT INQUIRE transaction 242  
 CEMT SET transaction 459

GTRAP  
 CSFE DEBUG 608

## H

hardcopy terminal 616

HBACKOUT  
 CEMT INQUIRE UOW 359

HCOMMIT  
 CEMT INQUIRE UOW 359

HDISABLED  
 CEMT INQUIRE URIMAP 377

HEADING  
 CMSG 571  
 HEADNG NOT YES/NO AT XXXX  
 CMSG 582

hexadecimal values 143

HFSFILE  
 CEMT INQUIRE DJAR 206  
 CEMT INQUIRE DOCTEMPLATE 209  
 CEMT INQUIRE URIMAP 378

HHHHHH  
 CSPG 610, 616

HHMM  
 CMSG 575, 585

HHMMSS  
 CEMT INQUIRE VTAM 381  
 CEMT SET VTAM 516

HIGH  
 CEMT INQUIRE DB2CONN 190, 196  
 CEMT SET DB2CONN 431  
 CEMT SET DB2ENTRY 437

HOST  
 CEMT INQUIRE CORBASERVER 184  
 CEMT INQUIRE transaction 244  
 CEMT INQUIRE UOWLINK 372  
 CEMT INQUIRE URIMAP 377  
 CEMT SET transaction 460

HOSTCODEPAGE  
 CEMT INQUIRE URIMAP 378

HTIME  
 CEMT INQUIRE TASK 314

HTTP  
 CEMT INQUIRE URIMAP 377

HTTPS  
CEMT INQUIRE URIMAP 377  
HTYPE  
CEMT INQUIRE TASK 314  
HVALUE  
CEMT INQUIRE TASK 314

## I

ID  
CMSG 571  
ID OPERAND GT 62 CHARS  
CMSG 583  
ID OPERAND MISSING  
CMSG 583  
ID PARAM MISSING OR NO )  
CMSG 583  
IDENTIFIER  
CEMT INQUIRE TASK 315  
identifiers  
resource, lists of 140  
identifying terminals 5  
IDLE  
CEMT INQUIRE DELETSHIPED 201  
CEMT SET DELETSHIPED 440  
IIOP  
CEMT SET WORKREQUEST 519  
IMMCLOSE  
CEMT INQUIRE IRC 250  
CEMT INQUIRE VTAM 381  
CEMT SET IRC 462  
CEMT SET VTAM 516  
IMMEDIATE  
CEBT PERFORM SHUTDOWN 88  
CEMT PERFORM SHUTDOWN 405  
IMMQUIESCED  
CEMT INQUIRE DSNNAME 220  
CEMT SET DSNNAME 447  
inactivating debugging profiles  
with the 3270 interface 40  
with the web interface 22  
INACTIVE  
CEMT INQUIRE SYSTEM 310  
INDEXRECFULL  
CEMT INQUIRE UOWDSNFAIL 363  
INDIRECT  
CEMT INQUIRE CONNECTION 175  
CEMT INQUIRE TDQUEUE 330  
INDOUBT  
CEMT INQUIRE TASK 314  
CEMT INQUIRE TDQUEUE 328  
CEMT INQUIRE TRANSACTION 340  
CEMT INQUIRE UOW 359  
CEMT INQUIRE UOWDSNFAIL 363  
INDOUBTMINS  
CEMT INQUIRE TASK 314  
CEMT INQUIRE TRANSACTION 340  
INDOUBTWAIT  
CEMT INQUIRE TASK 315  
CEMT INQUIRE TDQUEUE 328  
CEMT INQUIRE TRANSACTION 341

INFLIGHT  
CEMT INQUIRE UOW 359  
Information Display System 567  
INPUT  
CEMT INQUIRE TDQUEUE 329  
CMSG 581  
inquire about a resource 135  
INQUIRE keyword, CEMT 136  
INQUIRE TCPIP SERVICE 321, 322, 323  
INS  
CEOT 522  
INSERVICE  
CEMT INQUIRE CONNECTION 180  
CEMT INQUIRE NETNAME 273  
CEMT INQUIRE TERMINAL 336  
CEMT SET CONNECTION 420  
CEMT SET LINE 466  
CEMT SET NETNAME 474  
CEMT SET TERMINAL 504  
INTERFACE  
CEMT INQUIRE REQUESTMODEL 292  
interface to master terminal, programmable 144  
INTERNAL  
CEMT INQUIRE TASK 316  
interregion communication (IRC) 249, 462  
INTERVAL  
CEMT INQUIRE DELETSHIPED 201  
CEMT INQUIRE STATISTICS 296  
CEMT SET DELETSHIPED 440  
CEMT SET STATISTICS 484  
INTFACETYPE  
CEMT INQUIRE REQUESTMODEL 292  
INTRA  
CEMT INQUIRE TDQUEUE 330  
INTRTRACE  
CEBT INQUIRE INTRTRACE 84  
CEBT INQUIRE transaction 84  
CEBT SET transaction 95  
CEMT INQUIRE INTRTRACE 246  
CEMT INQUIRE transaction 246  
CEMT SET transaction 461  
INVALID  
CEMT INQUIRE DSNNAME 222  
INVALID LDC AT XXXXXXXX  
CMSG 583  
INVALID OPND AT XXXXX  
CMSG 583  
INVALID TBL SFX AT XXXXX  
CMSG 583  
INVLD OPERATR ID AT XXXXX  
CMSG 583  
INVLD SEPARATOR AT XXXXX  
CMSG 583  
INVLD TERMINL ID XXXX  
CMSG 583  
invoking a transaction 2  
invoking CEMT from an application program 144  
IOERROR  
CEMT INQUIRE UOWDSNFAIL 364  
IOTYPE  
CEMT INQUIRE TDQUEUE 329

IRBATCH  
 CEMT INQUIRE transaction 248  
 IRC  
 CEMT INQUIRE CONNECTION 175  
 CEMT INQUIRE IRC 249  
 CEMT INQUIRE transaction 249  
 CEMT INQUIRE UOWLINK 373  
 CEMT SET transaction 462  
 IRC (interregion communication)  
 CEMT INQUIRE transaction 249  
 CEMT SET transaction 462  
 ISOLATE  
 CEMT INQUIRE TRANSACTION 341  
 ISOLATEST  
 CEMT INQUIRE TRANSACTION 341

## J

JAVA  
 CEMT INQUIRE PROGRAM 289  
 JMODEL  
 CEMT DISCARD 149  
 CEMT INQUIRE JMODEL 252  
 CEMT transaction 251  
 JNDIPREFIX  
 CEMT INQUIRE CORBASERVER 184  
 JOURNALMODEL  
 CEDA 110  
 JOURNALNAME  
 CEMT DISCARD 149  
 CEMT INQUIRE JMODEL 252  
 CEMT INQUIRE JOURNALNAME 254  
 CEMT PERFORM STATISTICS 409  
 CEMT transaction 253  
 JVM  
 CEMT INQUIRE transaction 256  
 CEMT SET PROGRAM 480  
 JVMCLASS  
 CEMT INQUIRE PROGRAM 289  
 CEMT SET PROGRAM 480  
 JVMPOOL  
 CEMT INQUIRE transaction 259  
 CEMT PERFORM STATISTICS 409  
 CEMT SET transaction 464  
 JVMProfile  
 CEMT SET PROGRAM 480  
 JVMPROFILE  
 CEMT INQUIRE PROGRAM 289  
 CEMT PERFORM STATISTICS 409  
 JVMPROGRAM  
 CEMT PERFORM STATISTICS 409

## K

KEEPTIME  
 CEMT INQUIRE BRFACILITY 165  
 kernel exit routines, global 608  
 key, tab 141  
 KEYLENGTH  
 CEMT INQUIRE FILE 239  
 CEMT SET FILE 456

keyword, minimum abbreviation 3  
 KILL  
 CEKL SET TASK 132

## L

LANGUAGE 529  
 CEMT INQUIRE PROGRAM 289  
 Language Environment run-time options (CLER) 557  
 Language Environment/370 289  
 LASTUSEDINT  
 CEMT INQUIRE TSQUEUE 354  
 CEMT SET TSQUEUE 510  
 LCKSTRUCFULL  
 CEMT INQUIRE UOWDSNFAIL 364  
 LDC (logical device code) 576  
 LDC NOT 2 CHARS AT XXXXX  
 CMSG 583  
 LE/370 289  
 LE370  
 CEMT INQUIRE PROGRAM 289  
 LENGTH  
 CEMT INQUIRE PROGRAM 289  
 CEMT INQUIRE TSQUEUE 354  
 LINE  
 CEMT INQUIRE LINE 262  
 CEMT INQUIRE transaction 261  
 CEMT SET transaction 466  
 LINK  
 CEMT INQUIRE UOW 358  
 CEMT INQUIRE UOWLINK 372  
 LINKSYSNET  
 CEMT INQUIRE BRFACILITY 165  
 LINKSYSTEM  
 CEMT INQUIRE BRFACILITY 165  
 LIST  
 CEDA 111  
 List all profiles page  
 application debugging profile manager  
 web interface 17  
 List Debugging Profiles screen  
 application debugging profile manager  
 3270 interface 37  
 LISTENERPORT option  
 INQUIRE WORKREQUEST command 389  
 Lists  
 of resource identifiers 140  
 LOAD  
 CEMT INQUIRE FILE 239  
 CEMT SET FILE 456  
 LOADTYPE  
 CEMT INQUIRE FILE 239  
 LOCAL  
 CEMT INQUIRE MONITOR 268  
 CEMT INQUIRE NETNAME 271  
 CEMT INQUIRE SYDUMPCODE 303  
 CEMT INQUIRE TERMINAL 335  
 CEMT INQUIRE TRDUMPCODE 346  
 CEMT SET SYDUMPCODE 487  
 CEMT SET TRDUMPCODE 508

LOCATION  
     CEMT INQUIRE TSMODEL 348  
     CEMT INQUIRE TSQUEUE 354  
     CEMT INQUIRE URIMAP 379  
     CEMT SET URIMAP 513  
 LOCKING  
     CEMT SET FILE 457  
 LOGDEFER  
     CEMT INQUIRE SYSTEM 308  
     CEMT SET SYSTEM 491  
 logical device code (LDC) 576  
 logical unit destination 576  
 LOSTLOCKS  
     CEMT INQUIRE DSNAME 220  
 LOW  
     CEMT INQUIRE DB2CONN 190, 196  
     CEMT SET DB2CONN 431  
     CEMT SET DB2ENTRY 438  
 LSRPOOL  
     CEDA 111  
     CEMT PERFORM STATISTICS 409  
 LTPNAME  
     CEMT INQUIRE PARTNER 276  
 LU61  
     CEMT INQUIRE CONNECTION 178  
     CEMT INQUIRE UOWLINK 373

**M**

MAIN  
     CEMT INQUIRE TSMODEL 348  
     CEMT INQUIRE TSQUEUE 354  
 manager, application debugging profile 15  
 MANUAL  
     CEBT SET TAKEOVER 97  
 MAPSET  
     CEDA 111  
     CEMT INQUIRE PROGRAM 289  
 master terminal operator  
     duties 6  
     restriction 7  
     transaction, CEBT 79  
     transaction, CEKL 127  
     transaction, CEMT 135  
     types of terminal 7  
 master terminal, programmable interface to 144  
 MAXACTIVE  
     CEMT INQUIRE TCLASS 318  
     CEMT SET TCLASS 495  
 MAXDATALEN  
     CEMT SET TCPIPService 499  
 MAXIMUM  
     CEMT INQUIRE AUTOINSTALL 157  
     CEMT INQUIRE MODENAME 264  
     CEMT INQUIRE SYDUMPCODE 304  
     CEMT INQUIRE TRDUMPCODE 346  
     CEMT SET SYDUMPCODE 487  
     CEMT SET TRDUMPCODE 509  
 MAXITEMLEN  
     CEMT INQUIRE TSQUEUE 354

MAXJVMTCBS  
     CEMT INQUIRE DISPATCHER 203  
     CEMT SET DISPATCHER 442  
 MAXNUMRECS  
     CEMT INQUIRE FILE 239  
     CEMT SET FILE 457  
 MAXOPENTCBS  
     CEMT INQUIRE DISPATCHER 203  
     CEMT SET DISPATCHER 443  
 MAXREQS  
     CEMT SET AUTOINSTALL 413  
 MAXSOCKETS  
     CEMT INQUIRE TCPIP 319  
     CEMT SET TCPIP 497  
 MAXSSLTCBS  
     CEMT INQUIRE DISPATCHER 203  
     CEMT SET DISPATCHER 443  
 MAXTASKS  
     CEMT INQUIRE SYSTEM 308  
     CEMT SET SYSTEM 491  
 MAXXPTCBS  
     CEMT INQUIRE DISPATCHER 203  
     CEMT SET DISPATCHER 443  
 MEDIATYPE  
     CEMT INQUIRE URIMAP 378  
 MEMBER  
     CEMT INQUIRE TDQUEUE 329  
 MEMBERNAME  
     CEMT INQUIRE CONNECTION 177  
 MESSAGE HAS BEEN ROUTED  
     MSG 583  
 message-switching  
     continuation 565  
     displaying at terminal 612  
     errors 579  
     examples 586  
 message-switching response  
     indicator 579  
     response 579  
 message-switching transaction  
     multiple destinations 563  
     saved status 565  
     subsystems 566  
     terminal systems 566  
     transaction MSG 563  
 messages  
     +DATE INVLD 2330 to 0030 586  
     ALL ROUTE ENTRIES DELETED 580  
     CANCELED - TEMP STRG ERR N 580  
     canceling 564  
     chaining 612  
     chaining in partition 8  
     CLEAR/CANCL or CONT INPUT/ROUTE/MSG 581  
     CONTINUE INPUT 581  
     CONTINUE INPUT OR MSG 581  
     CONTINUE MSG 581  
     CONTINUE ROUTE OPERAND 581  
     DATE ALREADY PASSED 581  
     DATE INVALID AT xxxxxxxx 581  
     DATE NOT xx/xx AT xxxxx 582  
     DATE TOO FAR IN FUTURE 582

messages (continued)

delivery 566  
displaying 612  
ERRTERM INVALID AT xxxxx 582  
ERRTERM INVLD/UNSUPP xxxx 582  
HEADNG NOT YES/NO AT xxxx 582  
ID OPERAND GT 62 CHARS 583  
ID PARAM MISSING OR NO ) 583  
INVALID LDC AT xxxxxxxx 583  
INVALID OPND AT xxxxx 583  
INVALID TBL SFX AT xxxxx 583  
INVLD OPERATR ID AT xxxxx 583  
INVLD SEPARATOR AT xxxxx 583  
INVLD TERMINL ID xxxx 583  
IO OPERAND GT 62 CHARS 583  
LDC NOT 2 CHARS AT xxxxx 583  
MESSAGE HAS BEEN ROUTED 583  
MESSAGE HAS BEEN SENT 619  
MSG NOT SENT- NO RECEIVER 583  
NEED OPCLASS OR ROUTE 584  
NO INPUT - REENTER 584  
NO MESSAGE PROVIDED 584  
NO MSG STRT QUOT AT xxxxx 584  
NO TERM LIST TBL- xx 584  
NO TRMID FOR LDC AT xxxxx 584  
OPCLASS NOT 1-24 AT xxxxx 584  
OPERATORS NOT SIGNDON NNN 584  
OPRTR ID NOT SIGNDON /xxx 584  
PROTECT OPND NOT 1-2 CHAR 585  
purging 9  
ROUTE INVALID AT xxxxx 585  
sending 563  
SPACE MUST FOLLOW TRANSID 585  
TERMID GT 4 CHAR AT xxxxx 585  
TERMID NOT +/- AT xxxxx 585  
TERMINATED BY OPERATOR 585, 619  
terminating in partition 9  
termination 615  
TIME ALREADY PASSED 585  
TIME INVALID AT xxxxx 585  
TIME NOT 4 CHARS AT xxxxx 586  
TOO MANY TBL SFX AT xxxxx 586  
UNBALANCD PARENS AT xxxxx 586  
UNSUP OPRT TRMNL xxxxxxxx 586  
UNSUP TERMINAL xxxxxxxx 586  
ZERO LENGTH MSG AT " 586  
messages and codes, display 559  
MESSCHN  
  CSPG 613  
MINITEMLEN  
  CEMT INQUIRE TSQUEUE 354  
minus sign following CEMT 135  
MM/DD  
  CMSG 568, 570  
MM/DD/YY  
  CMSG 568  
MM/DD/YYYY  
  CMSG 570  
MOD  
  CEMT INQUIRE TDQUEUE 328

MODENAME  
  CEMT INQUIRE MODENAME 264  
  CEMT INQUIRE transaction 263  
  CEMT SET transaction 468  
MODENAME, CEMT requests 468  
MODIFY command 10  
MODULE  
  CEMT INQUIRE REQUESTMODEL 292  
MONITOR  
  CEMT INQUIRE MONITOR 266  
  CEMT INQUIRE transaction 265  
  CEMT PERFORM STATISTICS 409  
  CEMT SET transaction 470  
MRO (multiregion operation) 603  
MROBATCH  
  CEMT INQUIRE DISPATCHER 203  
  CEMT INQUIRE SYSTEM 308  
  CEMT SET DISPATCHER 443  
  CEMT SET SYSTEM 491  
MSG  
  CMSG 571, 581  
MSG NOT SENT - NO RECEIVER  
  CMSG 583  
MSGCOPY  
  CSPG 616  
MSGQUEUE1  
  CEMT INQUIRE DB2CONN 190  
  CEMT SET DB2CONN 430  
MSGQUEUE2  
  CEMT INQUIRE DB2CONN 190  
  CEMT SET DB2CONN 430  
MSGQUEUE3  
  CEMT INQUIRE DB2CONN 190  
  CEMT SET DB2CONN 430  
MSGTERM  
  CSPG 615  
multiple console support 9  
multiple destinations, message switching 563  
multiregion operation (MRO) 603  
MVS  
  CEMT INQUIRE JMODEL 252  
  CEMT INQUIRE JOURNALNAME 255  
  
**N**  
N906D  
  CEMT INQUIRE DB2CONN 192  
NAME  
  CEMT INQUIRE DOCTEMPLATE 209  
NAMEIND  
  CEMT INQUIRE TDQUEUE 329  
NAMESPACE  
  CEMT INQUIRE BRFACILITY 166  
NATLANG, system initialization parameter 530  
NATURE  
  CEMT INQUIRE NETNAME 271  
  CEMT INQUIRE TERMINAL 335  
NEED OPCLASS OR ROUTE  
  CMSG 584  
NET  
  CEOT 522

NETID  
  CEMT PERFORM ENDAFFINITY 401
NETNAME  
  CEMT INQUIRE BRFCAPILITY 166  
  CEMT INQUIRE CONNECTION 177  
  CEMT INQUIRE NETNAME 271  
  CEMT INQUIRE PARTNER 276  
  CEMT INQUIRE TERMINAL 335  
  CEMT INQUIRE transaction 269  
  CEMT INQUIRE UOW 358  
  CEMT INQUIRE UOWDSNFAIL 363  
  CEMT PERFORM ENDAFFINITY 401  
  CEMT SET transaction 473
NETUOWID  
  CEMT INQUIRE ENQ 227  
  CEMT INQUIRE UOW 358  
  CEMT INQUIRE UOWENQ 368  
  CEMT INQUIRE UOWLINK 373
NETWORK  
  CEMT INQUIRE PARTNER 276
networks  
  CEMT INQUIRE transaction 269  
  CEMT SET transaction 473
NEWCOPY  
  CEMT INQUIRE PROGRAM 287  
  CEMT SET PROGRAM 480
NEWPS 530
NEXT  
  CEBT INQUIRE AUXTRACE 80  
  CEBT SET AUXTRACE 91  
  CEMT INQUIRE AUXTRACE 160  
  CEMT SET AUXTRACE 414
NEXTTIME  
  CEMT INQUIRE STATISTICS 295, 297  
  CEMT SET STATISTICS 483
NO  
  MSG 571, 572
NO INPUT - REENTER  
  MSG 584
NO MESSAGE PROVIDED  
  MSG 584
NO MSG STRT QUOT AT XXXXX  
  MSG 584
NO TERM LIST TBL-XX  
  MSG 584
NO TRMID FOR LDC AT XXXXX  
  MSG 584
NOADDABLE  
  CEMT INQUIRE FILE 237  
  CEMT SET FILE 457
NOANALYZER  
  CEMT INQUIRE URIMAP 377
NOAPPLNAME  
  CEMT INQUIRE MONITOR 266
NOATI  
  CEMT INQUIRE NETNAME 270  
  CEMT INQUIRE TERMINAL 334  
  CEMT SET NETNAME 474  
  CEMT SET TERMINAL 504  
  CEOT 523
NOAUTO  
  CEMT SET CORBASERVER 425
NOAUTOSWITCH  
  CEBT INQUIRE DUMP 82  
  CEBT SET DUMP 93  
  CEMT INQUIRE DUMPDS 224  
  CEMT SET DUMPDS 451
NOBROWSE  
  CEMT INQUIRE FILE 237  
  CEMT SET FILE 457
NOCEDF  
  CEMT INQUIRE PROGRAM 287  
  CEMT SET PROGRAM 481
NOCMDPROT  
  CEMT INQUIRE SYSTEM 306
NOCONNECT  
  CEMT INQUIRE DB2CONN 191  
  CEMT SET DB2CONN 432
NOCONVERSE  
  CEMT INQUIRE MONITOR 266  
  CEMT SET MONITOR 471
NOCREATE  
  CEMT INQUIRE NETNAME 271  
  CEMT INQUIRE TERMINAL 334  
  CEMT SET NETNAME 474  
  CEMT SET TERMINAL 504
NODAE  
  CEMT INQUIRE SYDUMPCODE 303  
  CEMT SET SYDUMPCODE 487
NODEBUG  
  CEMT SET SYSTEM 491
NODELETE  
  CEMT INQUIRE FILE 238  
  CEMT SET FILE 457
NOEMPTYREQ  
  CEMT INQUIRE FILE 238  
  CEMT SET FILE 457
NOEXCEPT  
  CEMT INQUIRE MONITOR 266  
  CEMT SET MONITOR 471
NOEXCLUSIVE  
  CEMT INQUIRE FILE 239  
  CEMT SET FILE 457
NOFORCE  
  CEMT INQUIRE SYSTEM 307  
  CEMT SET SYSTEM 492
NOISOLATE  
  CEMT INQUIRE TRANSACTION 341
NOLOAD  
  CEMT INQUIRE FILE 239  
  CEMT SET FILE 457
NOLOSTLOCKS  
  CEMT INQUIRE DSNAME 220
NONCRITICAL value  
  INQUIRE TCPIP SERVICE 323
NONE  
  CEMT INQUIRE DB2CONN 187  
  CEMT INQUIRE DB2ENTRY 194  
  CEMT INQUIRE URIMAP 378  
  CEMT SET DB2CONN 427  
  CEMT SET DB2ENTRY 436

NONE (continued)  
 CEMT SET URIMAP 513  
 CSFE 608  
 NONTERMREL  
 CEMT INQUIRE DB2CONN 190  
 CEMT SET DB2CONN 430  
 NOPERF  
 CEMT INQUIRE MONITOR 267  
 CEMT SET MONITOR 471  
 NOREAD  
 CEMT INQUIRE FILE 241  
 CEMT SET FILE 457  
 NORECOVDATA  
 CEMT INQUIRE CONNECTION 179  
 CEMT SET CONNECTION 421  
 NOREENTPROTECT  
 CEMT INQUIRE SYSTEM 309  
 NORELEASE  
 CEMT INQUIRE DB2CONN 190  
 CEMT SET DB2CONN 430  
 NORESRCE  
 CEMT INQUIRE MONITOR 267  
 CEMT SET MONITOR 471  
 NORESTART  
 CEMT PERFORM SHUTDOWN 405  
 NORESYNC  
 CEMT INQUIRE DB2CONN 191  
 CEMT SET DB2CONN 431  
 NORETAINED  
 CEMT INQUIRE DSNAME 221  
 NORMI  
 CEMT INQUIRE MONITOR 267  
 NOROLLBACK  
 CEMT INQUIRE DB2CONN 189  
 CEMT INQUIRE DB2ENTRY 196  
 NOSDTRAN  
 CEMT PERFORM SHUTDOWN 405  
 NOSECURITY  
 CEMT INQUIRE TSMODEL 349  
 NOSHUTDOWN  
 CEMT INQUIRE SYDUMPCODE 304  
 CEMT INQUIRE TRDUMPCODE 346  
 CEMT SET SYDUMPCODE 487  
 CEMT SET TRDUMPCODE 509  
 NOSWITCH  
 CEMT INQUIRE AUXTRACE 80  
 CEMT SET AUXTRACE 91  
 CEMT INQUIRE AUXTRACE 160  
 CEMT SET AUXTRACE 414  
 NOSYNCPOINT  
 CEMT INQUIRE MONITOR 267  
 CEMT SET MONITOR 471  
 NOSYS DUMP  
 CEMT INQUIRE SYDUMPCODE 304  
 CEMT INQUIRE TRDUMPCODE 346  
 CEMT SET SYDUMPCODE 487  
 CEMT SET SYSTEM 492  
 CEMT SET TRDUMPCODE 509  
 NOSYSLOG  
 CEMT INQUIRE STREAMNAME 300  
 NOTAPPLIC  
 CEMT INQUIRE DSNAME 219  
 CEMT INQUIRE UOWDSNFAIL 364  
 INQUIRE TCPIP SERVICE 322  
 notation, syntax 2  
 NOTCONNECTED  
 CEMT INQUIRE DB2CONN 189  
 CEMT SET DB2CONN 429  
 NOTDEFINED  
 CEMT INQUIRE PROGRAM 289  
 NOTDYNAMIC  
 CEMT INQUIRE PROGRAM 288  
 NOTPENDING  
 CEMT INQUIRE CONNECTION 177  
 CEMT SET CONNECTION 421  
 NOTPURGEABLE  
 CEMT INQUIRE TRANSACTION 342  
 CEMT SET TRANSACTION 507  
 NOTRANDUMP  
 CEMT INQUIRE TRDUMPCODE 346  
 CEMT SET TRDUMPCODE 509  
 NOTRECOVERABLE  
 CEMT INQUIRE DSNAME 221  
 CEMT INQUIRE TSMODEL 348  
 CEMT INQUIRE TSQUEUE 355  
 NOTREQUIRED  
 CEMT INQUIRE PROGRAM 287  
 NOTRLS  
 CEMT INQUIRE FILE 241  
 CEMT INQUIRE UOWDSNFAIL 365  
 CEMT SET FILE 457  
 NOTROUTABLE  
 CEMT INQUIRE TRANSACTION 342  
 NOTSOS  
 CEMT INQUIRE DSAS 212  
 CEMT INQUIRE SYSTEM 310  
 NOTSUPPORTED  
 CEMT INQUIRE CONNECTION 176  
 NOTTABLE  
 CEMT INQUIRE FILE 241  
 CEMT SET FILE 458  
 NOTTI  
 CEMT INQUIRE NETNAME 274  
 CEMT INQUIRE TERMINAL 337  
 CEMT SET NETNAME 474  
 CEMT SET TERMINAL 504  
 CEOT 523  
 NOTWAIT  
 CEMT INQUIRE DB2CONN 192  
 CEMT INQUIRE DB2ENTRY 197  
 CEMT SET DB2CONN 433  
 CEMT SET DB2ENTRY 438  
 NOUPDATE  
 CEMT INQUIRE FILE 241  
 CEMT SET FILE 458  
 NOWAIT  
 CEMT INQUIRE TASK 315  
 CEMT INQUIRE TDQUEUE 328  
 CEMT INQUIRE TRANSACTION 341  
 NQNAME  
 CEMT INQUIRE CONNECTION 177

NQNAME (continued)

CEMT INQUIRE NETNAME 272  
CEMT INQUIRE TERMINAL 335

NRS

CEMT INQUIRE CONNECTION 179

NUMBER

CEMT INQUIRE LINE 262  
CEMT SET LINE 466

NUMITEMS

CEMT INQUIRE TSQUEUE 354

## O

OAPPLID option

INQUIRE WORKREQUEST command 389

OBJECT

CEMT INQUIRE DSNAME 220

OBTAINING

CEMT INQUIRE CONNECTION 176

OFF

CEBT SET SURVEILLANCE 96  
CEDF 121  
CEDX 125  
CEMT INQUIRE MONITOR 267  
CEMT INQUIRE STATISTICS 297  
CEMT SET MONITOR 472  
CEMT SET PROCESSTYPE 477  
CEMT SET STATISTICS 484  
CSFE SPOOLFE 606  
CSFE ZCQTRACE 606

OK

CEMT INQUIRE STREAMNAME 300  
CEMT INQUIRE UOWLINK 373

OLD

CEMT INQUIRE FILE 238  
CEMT INQUIRE TDQUEUE 328  
CEMT SET FILE 458

OLDCACHES

CEMT INQUIRE CLASSCACHE 171

ON

CEBT SET SURVEILLANCE 96  
CEDF 122  
CEDX 125  
CEMT INQUIRE MONITOR 267  
CEMT INQUIRE STATISTICS 297  
CEMT SET MONITOR 472  
CEMT SET STATISTICS 484  
CSFE SPOOLFE 606

OPCLASS

MSG 572

OPCLASS NOT 1-24 AT XXXXX

MSG 584

OPEN

CEBT INQUIRE DUMP 82  
CEBT SET DUMP 93  
CEMT INQUIRE DUMPDS 224  
CEMT INQUIRE FILE 240  
CEMT INQUIRE IRC 250  
CEMT INQUIRE RRMS 294  
CEMT INQUIRE TDQUEUE 329  
CEMT INQUIRE VTAM 381

OPEN (continued)

CEMT SET DUMPDS 451  
CEMT SET FILE 458  
CEMT SET IRC 462  
CEMT SET TDQUEUE 501  
CEMT SET VTAM 516

OPENAPI

CEMT INQUIRE PROGRAM 287

OPENERROR

CEMT INQUIRE UOWDSNFAIL 364

OPENSTATUS

CEMT INQUIRE DUMPDS 224  
CEMT INQUIRE FILE 240  
CEMT INQUIRE IRC 249  
CEMT INQUIRE RRMS 294  
CEMT INQUIRE TDQUEUE 329  
CEMT INQUIRE VTAM 381  
CEMT SET TCPIP 496  
CEMT SET TCPIPSERVICE 499

operating system console as a terminal 9

OPERATION

CEMT INQUIRE REQUESTMODEL 293

operator security 5

OPERATORS NOT SIGNDON NNN

MSG 584

OPID

CEMT INQUIRE DB2CONN 187  
CEMT INQUIRE DB2ENTRY 195  
CEMT SET DB2CONN 428  
CEMT SET DB2ENTRY 436  
MSG 573

OPRTR ID NOT SIGNDON /XXX

MSG 584

OSLEVEL

CEMT INQUIRE SYSTEM 308

OTASK option

INQUIRE WORKREQUEST command 389

OTRANSID option

INQUIRE WORKREQUEST command 389

OTSCoord

CEMT INQUIRE UOW 360

OTSTID

CEMT INQUIRE UOW 358

OTSTIMEOUT

CEMT INQUIRE TRANSACTION 341

OUTPUT

CEMT INQUIRE TDQUEUE 329

OUTSERVICE

CEMT INQUIRE CONNECTION 180  
CEMT INQUIRE NETNAME 273  
CEMT INQUIRE TERMINAL 336  
CEMT SET CONNECTION 422  
CEMT SET LINE 467  
CEMT SET NETNAME 474  
CEMT SET TERMINAL 504

overtyping a display 140

OWNER

CEMT INQUIRE ENQ 228  
CEMT INQUIRE UOWENQ 369

## P

- PA1 print key 8
- PAG
  - CEOT 522
- page copying 8, 616
- page retrieval
  - and partitions 8, 612
  - logical device 611
  - session 609
  - single keystroke 611
  - transaction, CSPG 609
- page retrieval function
  - new PROFILE definition, DFHCICSP 4
- PAGEABLE
  - CEMT INQUIRE NETNAME 272
  - CEMT INQUIRE TERMINAL 335
  - CEMT SET NETNAME 475
  - CEMT SET TERMINAL 505
  - CEOT 523
- PAGESTATUS
  - CEMT INQUIRE NETNAME 272
  - CEMT INQUIRE TERMINAL 335
- paging command 609
- PARAMETERS
  - CSPG 613
- partitions 612
- partitions, BMS 7
- PARTITIONSET
  - CEDA 111
  - CEMT INQUIRE PROGRAM 290
- PARTNER
  - CEDA 111
  - CEMT DISCARD 149
  - CEMT INQUIRE PARTNER 276
  - CEMT INQUIRE transaction 275
- PATH
  - CEMT INQUIRE DSNAME 220
  - CEMT INQUIRE URIMAP 377
- PAUSE
  - CEBT INQUIRE AUXTRACE 80
  - CEBT SET AUXTRACE 91
  - CEMT INQUIRE AUXTRACE 160
  - CEMT SET AUXTRACE 414
- PENDING
  - CEMT INQUIRE CONNECTION 177
- PENDSTATUS
  - CEMT INQUIRE CONNECTION 177
- PERF
  - CEMT INQUIRE MONITOR 267
  - CEMT SET MONITOR 472
- PERFCLASS
  - CEMT INQUIRE MONITOR 267
- PERFORM, CEBT
  - SHUTDOWN 88
  - SNAP 89
  - TAKEOVER 90
- PERFORM, CEMT
  - RESET 403
  - SECURITY 404
  - SHUTDOWN 405
  - STATISTICS 407
- PERMANENT
  - CEMT INQUIRE URIMAP 379
  - CEMT SET URIMAP 513
- PF (program function) key 138
- PGCHAIN, system initialization parameter 609, 613
- PGCOPY, system initialization parameter 609, 616
- PGMINTERFACE
  - CEMT INQUIRE WEBSERVICE 386
- PGPURGE, system initialization parameter 609, 615
- PGRET, system initialization parameter 609, 610, 612
- PHASEIN
  - CEMT INQUIRE PROGRAM 288
  - CEMT SET PROGRAM 481
- PHASEOUT
  - CEMT PERFORM CLASSCACHE command 393
  - CEMT SET JVMPOOL 464
- PHASINGOUT
  - CEMT INQUIRE CLASSCACHE 171
  - CEMT INQUIRE JVMPOOL 259
- PHASINGOUTST
  - CEMT INQUIRE JVM 257
- PIPELINE
  - CEDA 111
  - CEMT DISCARD 149
  - CEMT INQUIRE transaction 277
  - CEMT INQUIRE URIMAP 377, 378
  - CEMT INQUIRE WEBSERVICE 385
  - CEMT PERFORM PIPELINE 402
  - CEMT PERFORM STATISTICS 409
  - CEMT SET PIPELINE 476
  - CEMT transaction 402
- PLAN
  - CEMT INQUIRE DB2CONN 190
  - CEMT INQUIRE DB2ENTRY 196
  - CEMT SET DB2CONN 430
  - CEMT SET DB2ENTRY 437
- PLANEXITNAME
  - CEMT INQUIRE DB2CONN 190
  - CEMT INQUIRE DB2ENTRY 196
  - CEMT SET DB2CONN 430
  - CEMT SET DB2ENTRY 437
- PLI
  - CEMT INQUIRE PROGRAM 289
- PLT
  - CEMT PERFORM SHUTDOWN 405
- plus sign in scrolling under CEMT 141
- POOL
  - CEMT SET DB2ENTRY 437
- POOLNAME
  - CEMT INQUIRE TSMODEL 348
  - CEMT INQUIRE TSQUEUE 352
  - CEMT SET TSQUEUE 510
- positional fields 140
- PREFIX
  - CEMT INQUIRE TSMODEL 348
  - CMSG 572
- PRFILE
  - CEMT INQUIRE TRANSACTION 341
- PRGDLAY, system initialization parameter 570
- PRI
  - CEOT 522

PRINT  
   CSFE 605  
 print (PA1) key 8  
 PRIORITY  
   CEMT INQUIRE DB2CONN 190  
   CEMT INQUIRE DB2ENTRY 196  
   CEMT INQUIRE NETNAME 272  
   CEMT INQUIRE TASK 315  
   CEMT INQUIRE TERMINAL 335  
   CEMT INQUIRE TRANSACTION 341  
   CEMT SET DB2CONN 431  
   CEMT SET DB2ENTRY 437  
   CEMT SET NETNAME 475  
   CEMT SET TASK 494  
   CEMT SET TERMINAL 505  
   CEMT SET TRANSACTION 507  
 PRIVATE  
   CEMT INQUIRE PROGRAM 290  
   CEMT SET PROGRAM 481  
 PROCESS  
   CEMT INQUIRE TASK 315  
   CEMT SET PROCESSTYPE 477  
 PROCESSTYPE  
   CEDA 111  
   CEMT DISCARD 149  
   CEMT INQUIRE TASK 315  
   CEMT INQUIRE transaction 280  
   CEMT SET transaction 477  
 PROFILE  
   CEDA 111  
   CEMT DISCARD 150  
   CEMT INQUIRE CLASSCACHE 171  
   CEMT INQUIRE JVM 258  
   CEMT INQUIRE PARTNER 276  
   CEMT INQUIRE PROFILE 282  
   CEMT INQUIRE transaction 282  
   CEMT PERFORM CLASSCACHE command 394  
 profile manager, application debugging 15  
 profile manager, debugging  
   web interface 16  
   configuring 16  
 profile, rebuild 404  
 PROGAUTO  
   CEMT PERFORM STATISTICS 409  
 PROGAUTOCTLG  
   CEMT INQUIRE SYSTEM 308  
   CEMT SET SYSTEM 490  
 PROGAUTOEXIT  
   CEMT INQUIRE SYSTEM 308  
   CEMT SET SYSTEM 492  
 PROGAUTOINST  
   CEMT INQUIRE SYSTEM 308  
   CEMT SET SYSTEM 490  
 PROGRAM  
   CEDA 111  
   CEMT DISCARD 150  
   CEMT INQUIRE AUTOINSTALL 158  
   CEMT INQUIRE PROGRAM 289  
   CEMT INQUIRE transaction 283  
   CEMT INQUIRE TRANSACTION 341  
   CEMT INQUIRE URIMAP 378  
 PROGRAM (*continued*)  
   CEMT INQUIRE WEBSERVICE 386  
   CEMT PERFORM STATISTICS 409  
   CEMT SET AUTOINSTALL 413  
   CEMT SET PROGRAM 479  
   CEMT SET transaction 479  
 program function (PF) key 138  
 programmable interface to master terminal 144  
 PROGTYPE  
   CEMT INQUIRE PROGRAM 289  
 PROTECT  
   CMSG 572  
 PROTECT OPND NOT 1-2 CHAR  
   CMSG 585  
 PROTECTNUM  
   CEMT INQUIRE DB2ENTRY 196  
   CEMT SET DB2ENTRY 438  
 PROTOCOL  
   CEMT INQUIRE CONNECTION 178  
   CEMT INQUIRE UOWLINK 373  
 PS  
   CESN 530  
 PSDINTERVAL  
   CEMT INQUIRE VTAM 381  
   CEMT SET VTAM 516  
 pseudo-display fields 140  
 PTHREADS  
   CEMT INQUIRE DB2ENTRY 196  
 PUBLISH  
   CEMT PERFORM CORBASERVER 396  
   CEMT PERFORM DJAR 399  
 PURGE  
   CEKL SET TASK 132  
   CEMT INQUIRE CONNECTION 178  
   CEMT INQUIRE NETNAME 272  
   CEMT INQUIRE TASK 315  
   CEMT INQUIRE TERMINAL 336  
   CEMT PERFORM CLASSCACHE command 393  
   CEMT SET CONNECTION 423  
   CEMT SET JVMPOOL 464  
   CEMT SET LINE 467  
   CEMT SET NETNAME 475  
   CEMT SET TASK 494  
   CEMT SET TERMINAL 505  
   CEMT SET WORKREQUEST 520  
   INQUIRE WORKREQUEST command 389  
 PURGEABILITY  
   CEMT INQUIRE TRANSACTION 342  
 PURGEABLE  
   CEMT INQUIRE TRANSACTION 342  
   CEMT SET TRANSACTION 507  
 PURGECYCLEM  
   CEMT INQUIRE DB2CONN 190  
   CEMT SET DB2CONN 431  
 PURGECYCLES  
   CEMT INQUIRE DB2CONN 190  
   CEMT SET DB2CONN 431  
 PURGETHRESH  
   CEMT INQUIRE TCLASS 318  
   CEMT SET TCLASS 495

PURGETYPE  
 CEMT INQUIRE CONNECTION 178  
 CEMT INQUIRE NETNAME 272  
 CEMT INQUIRE TASK 315  
 CEMT INQUIRE TERMINAL 335  
 PURGETYPE option  
 INQUIRE WORKREQUEST command 389

## Q

Q  
 CSPG 612  
 QD  
 CEMT INQUIRE TASK 316  
 QLEN  
 CEMT INQUIRE ENQ 227  
 CEMT INQUIRE UOWENQ 368  
 QR  
 CEMT INQUIRE TASK 316  
 QUALIFIER  
 CEMT INQUIRE ENQ 227  
 CEMT INQUIRE UOWENQ 368  
 QUASIRENT  
 CEMT INQUIRE PROGRAM 287  
 question mark preceding CEMT 135  
 QUEUE  
 CEMT INQUIRE TDQUEUE 329  
 CEMT SET transaction 482  
 QUEUED  
 CEMT INQUIRE TCLASS 318  
 queues, CEMT requests 326, 482, 496, 500  
 QUIESCED  
 CEMT INQUIRE DSNAME 220  
 CEMT SET DSNAME 448  
 QUIESCESTATE  
 CEMT INQUIRE DSNAME 220

## R

RACF resource profile 404  
 RCONNECTION  
 CEMT INQUIRE CONNECTION 175  
 RDBACK  
 CEMT INQUIRE TDQUEUE 329  
 RDO (resource definition online) 103  
 RDSASIZE  
 CEMT INQUIRE DSAS 212  
 READ  
 CEMT INQUIRE FILE 240  
 CEMT SET FILE 458  
 READSTATUS  
 CEMT INQUIRE FILE 240  
 REASON  
 CEMT INQUIRE UOWDSNFAIL 363  
 REBUILD  
 CEMT PERFORM SECURITY 404  
 CEMT SET DB2CONN 432  
 rebuild resource profiles 404  
 RECONNECT  
 CEMT INQUIRE DB2CONN 191  
 CEMT SET DB2CONN 432

RECORD  
 CEMT INQUIRE STATISTICS 297  
 CEMT PERFORM STATISTICS 409  
 RECORDING  
 CEMT INQUIRE STATISTICS 297  
 RECORDNOW  
 CEMT SET STATISTICS 485  
 RECORDSIZE  
 CEMT INQUIRE FILE 241  
 CEMT SET FILE 458  
 RECOVDATA  
 CEMT INQUIRE CONNECTION 179  
 RECOVERABLE  
 CEMT INQUIRE DSNAME 221  
 CEMT INQUIRE TSMODEL 348  
 CEMT INQUIRE TSQUEUE 354  
 RECOVERED  
 CEMT INQUIRE DSNAME 217  
 CEMT SET DSNAME 448  
 RECOVERLOCKS  
 CEMT INQUIRE DSNAME 220  
 RECOVERY  
 CEMT PERFORM STATISTICS 409  
 RECOVSTATUS  
 CEMT INQUIRE CONNECTION 179  
 CEMT INQUIRE DSNAME 220  
 CEMT INQUIRE TSMODEL 348  
 CEMT INQUIRE TSQUEUE 354  
 REDIRECTTYPE  
 CEMT INQUIRE URIMAP 378  
 REENTPROTECT  
 CEMT INQUIRE SYSTEM 309  
 REGERROR  
 INQUIRE TCPIP SERVICE 323  
 REGISTERED  
 CEMT INQUIRE VTAM 381  
 INQUIRE TCPIP SERVICE 323  
 REJECT  
 CEMT INQUIRE TDQUEUE 329  
 RELATED  
 CEMT INQUIRE SYDUMPCODE 303  
 CEMT INQUIRE TRDUMPCODE 346  
 CEMT SET SYDUMPCODE 488  
 CEMT SET TRDUMPCODE 509  
 RELATION  
 CEMT INQUIRE ENQ 228  
 CEMT INQUIRE UOWENQ 369  
 RELEASE  
 CEMT INQUIRE DB2CONN 190  
 CEMT INQUIRE SYSTEM 309  
 CEMT SET DB2CONN 430  
 RELEASED  
 CEMT INQUIRE CONNECTION 176  
 CEMT INQUIRE NETNAME 273  
 CEMT INQUIRE TERMINAL 337  
 CEMT SET BR FACILITY 416  
 CEMT SET CONNECTION 423  
 CEMT SET NETNAME 475  
 CEMT SET TERMINAL 505  
 RELOAD  
 CEMT PERFORM CLASSCACHE command 394

REMLOSTLOCKS  
   CEMT INQUIRE DSNAME 220  
 REMOTE  
   CEMT INQUIRE FILE 237  
   CEMT INQUIRE TDQUEUE 330  
 REMOTEPREFIX  
   CEMT INQUIRE TSMODEL 348  
 REMOTESYSNET  
   CEMT INQUIRE BRFACILITY 166  
 REMOTESYSTEM  
   CEMT INQUIRE BRFACILITY 166  
   CEMT INQUIRE CONNECTION 179  
   CEMT INQUIRE NETNAME 272  
   CEMT INQUIRE PROGRAM 290  
   CEMT INQUIRE TERMINAL 336  
   CEMT INQUIRE TSMODEL 348  
 REMOVE  
   CEMT INQUIRE DSNAME 216  
   CEMT INQUIRE SYDUMPCODE 303  
   CEMT INQUIRE TRDUMPCODE 345  
   CEMT SET DSNAME 448  
   CEMT SET SYDUMPCODE 488  
   CEMT SET TRDUMPCODE 509  
 REPLY command 11  
 REQUESTID option  
   INQUIRE WORKREQUEST command 389  
 REQUESTMODEL  
   CEDA 111  
   CEMT DISCARD 150  
   CEMT PERFORM STATISTICS 409  
   CEMT transaction 291  
 REQUIRED  
   CEMT INQUIRE PROGRAM 287  
 RESCOUNT  
   CEMT INQUIRE PROGRAM 290  
 RESET  
   CEMT INQUIRE JOURNALNAME 254  
   CEMT INQUIRE STATISTICS 298  
   CEMT INQUIRE SYDUMPCODE 303  
   CEMT INQUIRE TRDUMPCODE 345  
   CEMT PERFORM RESET 403  
   CEMT SET JOURNALNAME 463  
   CEMT SET SYDUMPCODE 488  
   CEMT SET TRDUMPCODE 509  
 RESETLOCKS  
   CEMT INQUIRE DSNAME 217  
   CEMT SET DSNAME 449  
 RESETNOW  
   CEMT PERFORM STATISTICS 409  
   CEMT SET STATISTICS 485  
 RESOURCE  
   CEMT INQUIRE ENQ 228  
   CEMT INQUIRE UOWENQ 369  
 resource definition online transaction, CEDA 103  
 resource definitions  
   profile, DFHCICSP 4  
 resource groups  
   Subdefining 140  
 resource identifiers  
   lists of 140  
 RESRCE  
   CEMT INQUIRE MONITOR 267  
   CEMT SET MONITOR 472  
 RESRCECLASS  
   CEMT INQUIRE MONITOR 267  
 RESYNC  
   CEMT INQUIRE CONNECTION 180  
   CEMT INQUIRE DB2CONN 191  
   CEMT SET CONNECTION 423  
   CEMT SET DB2CONN 431  
 RESYNCMEMBER  
   CEMT INQUIRE DB2CONN 191  
   CEMT SET DB2CONN 431  
 RESYNCSTATUS  
   CEMT INQUIRE UOWLINK 373  
 RETAINED  
   CEMT INQUIRE DSNAME 221  
   CEMT INQUIRE ENQ 228  
   CEMT INQUIRE UOWENQ 369  
 RETLOCKS  
   CEMT INQUIRE DSNAME 221  
 RETRACT  
   CEMT PERFORM CORBASERVER 397  
   CEMT PERFORM DJAR 399  
 retrieve a page 8, 609  
 RETRY  
   CEMT INQUIRE DSNAME 218  
   CEMT SET DSNAME 449  
 REUSEST  
   CEMT INQUIRE CLASSCACHE 171  
   CEMT INQUIRE JVM 258  
 RLEN  
   CEMT INQUIRE ENQ 228  
   CEMT INQUIRE UOWENQ 369  
 rls  
   CEMT INQUIRE UOWDSNFAIL 365  
 RLS  
   CEMT INQUIRE FILE 241  
   CEMT SET FILE 458  
 RLSACCESS  
   CEMT INQUIRE FILE 241  
   CEMT INQUIRE UOWDSNFAIL 365  
 RLSGONE  
   CEMT INQUIRE UOWDSNFAIL 364  
 RLSSERVER  
   CEMT INQUIRE UOW 360  
   CEMT INQUIRE UOWDSNFAIL 363  
 RMI  
   CEMT INQUIRE MONITOR 267  
   CEMT INQUIRE UOWLINK 373  
 RMIQFY  
   CEMT INQUIRE UOWLINK 373  
 RMIST  
   CEMT INQUIRE MONITOR 267  
 RNAME 180  
   CEMT INQUIRE NETNAME 273  
   CEMT INQUIRE TERMINAL 336  
 RNETNAME 180  
   CEMT INQUIRE NETNAME 273  
   CEMT INQUIRE TERMINAL 336

ROLE  
     CEMT INQUIRE UOWLINK 373  
 ROLLBACK  
     CEMT INQUIRE DB2CONN 189  
     CEMT INQUIRE DB2ENTRY 196  
 ROUTABLE  
     CEMT INQUIRE TRANSACTION 342  
 ROUTE  
     CMSG 572, 581  
 ROUTE INVALID AT XXXXX  
     CMSG 585  
 routed pages 617  
 routing transaction, CRTE 603  
 ROUTSTATUS  
     CEMT INQUIRE TRANSACTION 342  
 RRCOMMITFAIL  
     CEMT INQUIRE UOWDSNFAIL 364  
 RRINDOUBT  
     CEMT INQUIRE UOWDSNFAIL 364  
 RRMS  
     CEMT INQUIRE transaction 294  
     CEMT INQUIRE UOW 360  
     CEMT INQUIRE UOWLINK 373  
 RSESSION  
     CEMT INQUIRE NETNAME 271  
     CEMT INQUIRE TERMINAL 335  
 RTERMINAL  
     CEMT INQUIRE NETNAME 271  
     CEMT INQUIRE TERMINAL 335  
 RUNAWAY  
     CEMT INQUIRE DISPATCHER 203  
     CEMT INQUIRE SYSTEM 309  
     CEMT SET DISPATCHER 443  
     CEMT SET SYSTEM 492  
 RUNNING  
     CEKL INQUIRE TASK 129  
     CEMT INQUIRE TASK 315  
 RUNSTATUS  
     CEMT INQUIRE TASK 315  
 RUNTIME  
     CEMT INQUIRE PROGRAM 290

## S

S  
     CEMT INQUIRE TASK 316  
 saved status, message switching 565  
 SCAN  
     CEMT PERFORM CORBASERVER 397  
     CEMT PERFORM PIPELINE 402  
 SCANDELAY  
     CEMT INQUIRE DISPATCHER 204  
     CEMT INQUIRE SYSTEM 309  
     CEMT SET DISPATCHER 443  
     CEMT SET SYSTEM 492  
 SCHEME  
     CEMT INQUIRE URIMAP 377  
 scrolling a display 141  
 SD  
     CEMT INQUIRE TASK 316

SDSASIZE  
     CEMT INQUIRE DSAS 212  
 SDTRAN  
     CEMT INQUIRE SYSTEM 309  
     CEMT PERFORM SHUTDOWN 405  
 SECURITY 404  
     CEMT INQUIRE DB2CONN 191  
     CEMT INQUIRE TSMODEL 349  
     CEMT SET DB2CONN 431  
 security checking 114, 144  
 security, operator 5  
 SECURITYST  
     CEMT INQUIRE TSMODEL 348  
 SEND  
     CMSG 575  
 sequential terminal 567  
 SERVER  
     CEMT INQUIRE URIMAP 376  
 SERVSTATUS  
     CEMT INQUIRE CONNECTION 180  
     CEMT INQUIRE LINE 262  
     CEMT INQUIRE NETNAME 273  
     CEMT INQUIRE TERMINAL 336  
 SESSBEANTIME  
     CEMT INQUIRE CORBASERVER 185  
     CEMT SET CORBASERVER 425  
 SESSION  
     CEMT INQUIRE NETNAME 271  
     CEMT INQUIRE TERMINAL 335  
 SESSIONID  
     CEDF 122  
 SESSIONS  
     CEDA 111  
 Set Compiled Debugging Display Device screen  
     application debugging profile manager  
         3270 interface 54  
 Set compiled display device page  
     application debugging profile manager  
         web interface 34  
 setting the display device  
     with the 3270 interface 53  
     with the web interface 33  
 SHARE  
     CEMT INQUIRE FILE 238  
     CEMT SET FILE 458  
 SHARED  
     CEMT INQUIRE PROGRAM 290  
     CEMT SET PROGRAM 481  
 SHARESTATUS  
     CEMT INQUIRE PROGRAM 290  
 SHELF  
     CEMT INQUIRE CORBASERVER 185  
     CEMT INQUIRE PIPELINE 279  
 shipped terminal definitions  
     deletion of  
         CEMT DJAR command 399  
         CEMT INQUIRE DELETSHIPED 200  
         CEMT PERFORM CORBASERVER  
             command 396  
         CEMT PERFORM DELETSHIPED  
             command 398

shipped terminal definitions (*continued*)  
 deletion of (*continued*)  
 CEMT SET DELETSHIPPED command 440

SHR  
 CEMT INQUIRE TDQUEUE 328

SHUNTED  
 CEMT INQUIRE UOW 360

SHUTDOWN  
 CEMT INQUIRE SYDUMPCODE 304  
 CEMT INQUIRE TRDUMPCODE 346  
 CEMT SET SYDUMPCODE 488  
 CEMT SET TRDUMPCODE 509

shutdown, CICS  
 CEBT 88  
 CEMT 405

SHUTOPTION  
 CEMT INQUIRE SYDUMPCODE 304  
 CEMT INQUIRE TRDUMPCODE 346

SIGN  
 CEMT INQUIRE DB2ENTRY 195  
 CEMT SET DB2CONN 427  
 CEMT SET DB2ENTRY 436

sign-off, CESF  
 fixed-format terminal 525

SIGNID  
 CEMT INQUIRE DB2CONN 187, 191  
 CEMT SET DB2CONN 432

signon, CESN  
 3270 screen 528  
 non-3270 screen 529

single keystroke retrieval (SKR) 611

SKR (single keystroke retrieval) 611

SKRxxxx, system initialization parameter 611

SMF  
 CEMT INQUIRE JMODEL 252  
 CEMT INQUIRE JOURNALNAME 255

SNAP  
 CEMT PERFORM DUMP 400  
 CEMT transaction 400

SNAP command  
 CEBT transaction 89

SOAP  
 CEMT SET WORKREQUEST 520

SOS  
 CEMT INQUIRE DSAS 212  
 CEMT INQUIRE SYSTEM 310

SOSABOVE  
 CEMT INQUIRE DSAS 213  
 CEMT INQUIRE SYSTEM 310

SOSBELOW  
 CEMT INQUIRE DSAS 213  
 CEMT INQUIRE SYSTEM 310

SOSSTATUS  
 CEMT INQUIRE DSAS 212  
 CEMT INQUIRE SYSTEM 310

SPACE MUST FOLLOW TRANSID  
 CMSG 585

SPOOLFE  
 CSFE 606

spooling interface trace 606

SQLCODE  
 CEMT INQUIRE DB2CONN 189  
 CEMT INQUIRE DB2ENTRY 196  
 CEMT SET DB2CONN 429  
 CEMT SET DB2ENTRY 437

SSLUNAUTH  
 CEMT INQUIRE CORBASERVER 185

STACK option  
 INQUIRE WORKREQUEST command 389

STANDBYMODE  
 CEMT INQUIRE DB2CONN 191  
 CEMT SET DB2CONN 432

START  
 CEBT INQUIRE AUXTRACE 80  
 CEBT INQUIRE GTFTRACE 83  
 CEBT INQUIRE INTTRACE 84  
 CEBT SET AUXTRACE 92  
 CEBT SET GTFTRACE 94  
 CEBT SET INTTRACE 95  
 CEMT INQUIRE AUXTRACE 160  
 CEMT INQUIRE GTFTRACE 243  
 CEMT INQUIRE INTTRACE 247  
 CEMT PERFORM CLASSCACHE command 394  
 CEMT SET AUXTRACE 414  
 CEMT SET GTFTRACE 459  
 CEMT SET INTTRACE 461

STARTCODE  
 CEMT INQUIRE TASK 315

STARTING  
 CEMT INQUIRE UOWLINK 373

starting a transaction 2

STATE  
 CEMT INQUIRE DJAR 206  
 CEMT INQUIRE ENQ 228  
 CEMT INQUIRE UOWENQ 369

STATIC  
 CEMT INQUIRE DSNAME 218

STATISTICS  
 CEMT INQUIRE STATISTICS 298  
 CEMT INQUIRE transaction 295  
 CEMT PERFORM 407  
 CEMT SET transaction 483

STATS  
 CEMT PERFORM STATISTICS 409

STATSQUEUE  
 CEMT INQUIRE DB2CONN 191  
 CEMT SET DB2CONN 432

status  
 hardcopy terminal 616  
 indicator 579

STATUS  
 CEMT INQUIRE AUXTRACE 160  
 CEMT INQUIRE CLASSCACHE 171  
 CEMT INQUIRE ENQMODEL 231  
 CEMT INQUIRE GTFTRACE 243, 246  
 CEMT INQUIRE JOURNALNAME 254  
 CEMT INQUIRE JVMPOOL 259  
 CEMT INQUIRE MONITOR 267  
 CEMT INQUIRE PROGRAM 290  
 CEMT INQUIRE STREAMNAME 300  
 CEMT INQUIRE TRANSACTION 342

STGFRZ  
   CSFE 607  
 STOP  
   CEBT INQUIRE AUXTRACE 81  
   CEBT SET AUXTRACE 92  
   CEBT SET GTFTRACE 94  
   CEBT SET INTTRACE 95  
   CEMT INQUIRE AUXTRACE 160  
   CEMT INQUIRE GTFTRACE 243  
   CEMT INQUIRE INTTRACE 247  
   CEMT SET AUXTRACE 415  
   CEMT SET GTFTRACE 459  
   CEMT SET INTTRACE 461  
 STOPPED  
   CEBT INQUIRE GTFTRACE 83  
   CEBT INQUIRE INTTRACE 84  
 STORAGE  
   CEMT PERFORM STATISTICS 410  
 storage freeze, CSFE 606  
 storage subpool 607  
 storage violation trap 607  
 STOREPROTECT  
   CEMT INQUIRE SYSTEM 310  
 STREAMNAME  
   CEMT INQUIRE JMODEL 252  
   CEMT INQUIRE JOURNALNAME 255  
   CEMT INQUIRE STREAMNAME 300  
   CEMT PERFORM STATISTICS 410  
   CEMT transaction 299  
 Subdefining  
   resource groups 140  
 SUBORDINATE  
   CEMT INQUIRE UOWLINK 373  
 subsystem message-switching 566  
 SUBSYSTEMID  
   CEMT INQUIRE MONITOR 267  
 SUBTASKS  
   CEMT INQUIRE DISPATCHER 204  
 supervisory terminal operator  
   duties 5  
   identifying terminals 5  
   SUPRID option 5  
   transaction CEST 531  
 SURVEILLANCE  
   CEBT transaction 85, 96  
 SUSPENDED  
   CEKL INQUIRE TASK 129  
   CEMT INQUIRE TASK 315  
 SWITCH  
   CEBT INQUIRE AUXTRACE 81  
   CEBT INQUIRE DUMP 82  
   CEBT SET AUXTRACE 92  
   CEBT SET DUMP 93  
   CEMT INQUIRE AUXTRACE 160  
   CEMT INQUIRE DUMPDS 224  
   CEMT SET AUXTRACE 415  
   CEMT SET DUMPDS 451  
 SWITCHACTION  
   CEMT INQUIRE AUXTRACE 160  
 SWITCHSTATUS  
   CEMT INQUIRE AUXTRACE 160  
   SWITCHSTATUS (continued)  
     CEMT INQUIRE DUMPDS 224  
 SYDUMPCODE  
   CEMT INQUIRE SYDUMPCODE 304  
   CEMT SET transaction 486  
   CEMT transaction 301  
 SYNCPOINT  
   CEMT INQUIRE MONITOR 267  
   CEMT SET MONITOR 472  
 SYNCPOINTST  
   CEMT INQUIRE MONITOR 267  
 syntax notation 2  
 SYSDUMP  
   CEMT INQUIRE SYDUMPCODE 304  
   CEMT INQUIRE TRDUMPCODE 346  
   CEMT PERFORM STATISTICS 410  
   CEMT SET SYDUMPCODE 488  
   CEMT SET SYSTEM 492  
   CEMT SET TRDUMPCODE 509  
 SYSDUMPING  
   CEMT INQUIRE SYDUMPCODE 304  
   CEMT INQUIRE TRDUMPCODE 346  
 SYSID  
   CEDF 122  
   CEMT INQUIRE TSQUEUE 352  
   CEMT INQUIRE UOW 359  
   CEMT INQUIRE UOWDSNFAIL 365  
   CEMT INQUIRE UOWLINK 373  
   CEMT SET TSQUEUE 510  
   CRTE transaction 603  
 SYSLOG  
   CEMT INQUIRE STREAMNAME 300  
 SYSTEM  
   CEMT INQUIRE transaction 305  
   CEMT SET transaction 489  
 system spooling interface trace 606  
 SYSTEMLOG  
   CEMT INQUIRE STREAMNAME 300

## T

tab key 141  
 TABLE  
   CEMT INQUIRE FILE 241  
 TABLEMGR  
   CEMT PERFORM STATISTICS 410  
 TABLENAME  
   CEMT INQUIRE FILE 241  
   CEMT SET FILE 458  
 TAKEOVER  
   CEBT 86  
   CEBT transaction 90, 97  
   CEMT PERFORM SHUTDOWN 405  
 task  
   CEKL INQUIRE TASK 128  
 Task  
   CEKL SET TASK 132  
 TASK  
   CEKL SET transaction 132  
   CEMT INQUIRE DB2CONN 187  
   CEMT INQUIRE DB2ENTRY 194

TASK (*continued*)

- CEMT INQUIRE EXCI 233
- CEMT INQUIRE JVM 258
- CEMT INQUIRE NETNAME 273
- CEMT INQUIRE TASK 314, 316
- CEMT INQUIRE TERMINAL 336
- CEMT SET DB2CONN 427
- CEMT SET DB2ENTRY 435
- CEMT SET transaction 493

task associated trap 607

TASK command

- CEKL INQUIRE transaction 128
- CEMT INQUIRE transaction 311

TASK option

- INQUIRE WORKREQUEST command 390

TASKDATAKEY

- CEMT INQUIRE TRANSACTION 342

TASKDATALOC

- CEMT INQUIRE TRANSACTION 343

TASKID

- CEMT INQUIRE BRFACILITY 166
- CEMT INQUIRE ENQ 228
- CEMT INQUIRE UOW 359
- CEMT INQUIRE UOWENQ 369

TASKREQ

- CSFE 607

tasks

- CEKL INQUIRE requests 128
- CEKL SET requests 132
- CEMT INQUIRE requests 311
- CEMT SET requests 493

TCB

- CEMT INQUIRE TASK 316

TCBLIMIT

- CEMT INQUIRE DB2CONN 192
- CEMT SET DB2CONN 432

TCBs

- CEMT INQUIRE DB2CONN 191

TCLASS

- CEMT DISCARD 150
- CEMT INQUIRE TASK 311
- CEMT INQUIRE TCLASS 318
- CEMT INQUIRE transaction 317
- CEMT INQUIRE TRANSACTION 343
- CEMT PERFORM STATISTICS 410
- CEMT SET transaction 495
- CEMT SET TRANSACTION 507

TCP/IP services, CEMT requests 319, 320

TCPIP

- CEMT INQUIRE transaction 319
- CEMT PERFORM STATISTICS 410
- CEMT SET transaction 496

TCPIP, CEMT requests 498

TCPIPservice

- CEMT PERFORM STATISTICS 410

TCPIPSERVICE

- CEDA 111
- CEMT DISCARD 150
- CEMT INQUIRE HOST 245
- CEMT INQUIRE transaction 320
- CEMT INQUIRE URIMAP 377

TCPIPSERVICE (*continued*)

- CEMT SET transaction 498

TDQ

- CEMT INQUIRE ENQ 228
- CEMT INQUIRE UOWENQ 369

TDQUEUE

- CEDA 111
- CEMT DISCARD 150
- CEMT INQUIRE TDQUEUE 330
- CEMT INQUIRE transaction 326
- CEMT PERFORM STATISTICS 410
- CEMT SET transaction 500

TEMPLATENAME

- CEMT INQUIRE DOCTEMPLATE 209
- CEMT INQUIRE URIMAP 378

TEMPLATETYPE

- CEMT INQUIRE DOCTEMPLATE 209

TEMPORARY

- CEMT INQUIRE URIMAP 378
- CEMT SET URIMAP 513

temporary storage browse transaction, CEBR 75

TER

- CEOT 522

TERM

- CEMT INQUIRE DB2CONN 187
- CEMT INQUIRE DB2ENTRY 195
- CEMT INQUIRE TASK 314
- CEMT SET DB2CONN 428
- CEMT SET DB2ENTRY 436

TERMID

- CEDF 122
- CEMT INQUIRE BRFACILITY 166
- CEMT INQUIRE TDQUEUE 330
- CEMT INQUIRE UOW 359
- CMSG 573
- CSFE ZCQTRACE 606

TERMID GT 4 CHAR AT XXXXX

- CMSG 585

TERMID NOT +/- AT xxxxx

- CMSG 585

TERMID/OPID

- CMSG 573

TERMID\*LDC

- CMSG 577

TERMINAL

- CEDA 111
- CEMT DISCARD 150
- CEMT INQUIRE NETNAME 273
- CEMT INQUIRE TERMINAL 336
- CEMT INQUIRE transaction 331
- CEMT PERFORM STATISTICS 410
- CEMT SET transaction 502

terminal and system test transaction, CSFE 605

terminal builder trace 606

terminal list table (TLT) 5

terminal operator

- duties 5
- master 6
- paging 609
- supervisory 5
- transactions 4

terminal operator (*continued*)  
 write to console operator (CWTO) 619

terminals  
 3270 displays 567  
 3767 keyboard/prINTER 567  
 autopaging 616  
 CEMT INQUIRE requests 331  
 CEMT SET requests 502  
 console as a CICS terminal 9  
 fixed-format 525  
 identifying 5  
 message-switching transaction 566  
 paging 8, 609  
 sequential 567  
 status 521  
 test 605

terminate a logical message 9, 615

TERMINATED BY OPERATOR  
 CMSG 585

TERMLIST  
 CMSG 574

TERMLIST\*LDC  
 CMSG 577

TERMSTATUS  
 CEMT INQUIRE BR FACILITY 166  
 CEMT INQUIRE NETNAME 273  
 CEMT INQUIRE TERMINAL 337

testing a terminal 605

THREADERROR  
 CEMT INQUIRE DB2CONN 192

THREADLIMIT  
 CEMT INQUIRE DB2CONN 192  
 CEMT INQUIRE DB2ENTRY 196  
 CEMT SET DB2CONN 432  
 CEMT SET DB2ENTRY 438

THREADS  
 CEMT INQUIRE DB2CONN 192  
 CEMT INQUIRE DB2ENTRY 196

THREADSAFE  
 CEMT INQUIRE PROGRAM 287

THREADWAIT  
 CEMT INQUIRE DB2CONN 192  
 CEMT INQUIRE DB2ENTRY 197  
 CEMT SET DB2CONN 432  
 CEMT SET DB2ENTRY 438

TIME  
 CEMT INQUIRE DISPATCHER 204  
 CEMT INQUIRE MONITOR 267  
 CEMT INQUIRE SYSTEM 310  
 CEMT SET DISPATCHER 444  
 CEMT SET SYSTEM 492  
 CMSG 575

TIME ALREADY PASSED  
 CMSG 585

TIME INVALID AT XXXXX  
 CMSG 585

TIME NOT 4 CHARS AT XXXXX  
 CMSG 586

time-of-day resetting 403

TIMEOUTINT  
 CEMT INQUIRE WEB 382

TIMEOUTINT (*continued*)  
 CEMT SET WEB 517

TIMESTAMP  
 CEMT INQUIRE DJAR 207

TIMESTARTED  
 CEMT INQUIRE CLASSCACHE 172

TITLE  
 CEMT PERFORM DUMP 400

TLT (terminal list table) 5

TO  
 CEDA 111  
 CEMT INQUIRE TASK 316

TOO MANY TBL SFX AT XXXXX  
 CMSG 586

TOTAL  
 CEMT INQUIRE JVMPOOL 259

TOTALJVMS  
 CEMT INQUIRE CLASSCACHE 172

TP  
 CEMT INQUIRE TASK 316

TPNAME  
 CEMT INQUIRE PARTNER 276

TPOOL  
 CEMT INQUIRE DB2ENTRY 197  
 CEMT SET DB2ENTRY 438

TRA  
 CEOT 522

trace control transaction 533  
 trace program exit routine 608

TRANClass  
 CEKL INQUIRE TASK 128

TRANCLASS  
 CEDA 112

TRANDUMP  
 CEMT INQUIRE TRDUMPCODE 346  
 CEMT SET TRDUMPCODE 509

TRANDUMPING  
 CEMT INQUIRE TRDUMPCODE 346

TRANID  
 CEDX 125  
 CEMT INQUIRE TASK 316  
 CEMT INQUIRE TDQUEUE 330  
 CSFE 607

TRANISOLATE  
 CEMT INQUIRE SYSTEM 310

TRANSACTION  
 CEDA 112  
 CEMT DISCARD 150  
 CEMT INQUIRE BR FACILITY 166  
 CEMT INQUIRE NETNAME 274  
 CEMT INQUIRE TERMINAL 337  
 CEMT INQUIRE transaction 338  
 CEMT INQUIRE TRANSACTION 343  
 CEMT INQUIRE URIMAP 377  
 CEMT PERFORM STATISTICS 410  
 CEMT SET transaction 506

transaction resynchronization, CPIA 589  
 transaction routing, CRTE 603  
 transaction, BTS browser 57  
 transaction, in-doubt test tool 553  
 transaction, trace control 533

- transactions
  - all (list) 623
  - canceling 2
  - CBAM 57
  - CCRL 59
  - CEBT 79
  - CECI 99
  - CECS 99, 101
  - CEDA 103
  - CEDB 103
  - CEDC 103
  - CEDF 121
  - CEKL 127
  - CEMT 135
  - CEOT 521
  - CESN 527
  - CEST 531
  - CMSG 563
  - CPIA 589
  - CREA 591
  - CREC 591
  - CRTE 603
  - CSFE 605
  - CSPG 609
  - CWTO 619
  - DSNC 621
  - identification codes 5
  - initiating from console 10
  - invocation of 2
  - security key 5
  - transactions with operator interface 623
- transactions with operator interface
  - CADP 37
  - CBAM 57
  - CDBC 61
  - CDBI 65
  - CDBM 67
  - CESF 525
  - CETR 533
  - CIDP 551
  - CLER 557
  - CMAC 559
- TRANSID
  - CEKL INQUIRE TASK 129
  - CEMT INQUIRE DB2TRAN 199
  - CEMT INQUIRE ENQ 228
  - CEMT INQUIRE REQUESTMODEL 293
  - CEMT INQUIRE TSQUEUE 355
  - CEMT INQUIRE UOW 359
  - CEMT INQUIRE UOWENQ 369
  - CEMT SET DB2TRAN 439
- TRANSID option
  - INQUIRE WORKREQUEST command 390
- TRAP
  - CSFE DEBUG 608
- trap program exit routine 608
- traps
  - storage violation 607
  - task associated 607
- TRDUMP
  - CEMT PERFORM STATISTICS 410
- TRDUMPCODE
  - CEMT INQUIRE transaction 344
  - CEMT INQUIRE TRDUMPCODE 346
  - CEMT SET transaction 508
- TRIGGERLEVEL
  - CEMT INQUIRE TDQUEUE 330
  - CEMT SET TDQUEUE 501
- TRPROF
  - CEMT INQUIRE TRANSACTION 343
  - CRTE transaction 603
- TSMODEL
  - CEDA 112
  - CEMT DISCARD 150
  - CEMT INQUIRE transaction 347
  - CEMT INQUIRE TSMODEL 349
- TSO console as a CICS terminal 12
- TSPOOL
  - CEMT INQUIRE transaction 350
  - CEMT INQUIRE TSPOOL 351
- TSQ
  - CEMT INQUIRE ENQ 229
  - CEMT INQUIRE UOWENQ 370
- TSQNAME
  - CEMT INQUIRE transaction 352
- TSQUEUE
  - CEMT INQUIRE transaction 352
  - CEMT INQUIRE TSQUEUE 355
  - CEMT PERFORM STATISTICS 410
  - CEMT SET transaction 510
  - CEMT SET TSQUEUE 510
- TSQUEUELIMIT
  - CEMT INQUIRE MONITOR 268
- TSYSTEM option
  - INQUIRE WORKREQUEST command 390
- TTI
  - CEMT INQUIRE NETNAME 274
  - CEMT INQUIRE TERMINAL 337
  - CEMT SET NETNAME 475
  - CEMT SET TERMINAL 505
  - CEOT 522, 523
- TTISTATUS
  - CEMT INQUIRE NETNAME 274
  - CEMT INQUIRE TERMINAL 337
- TTTT
  - CSPG 616
- TWAIT
  - CEMT INQUIRE DB2CONN 192
  - CEMT INQUIRE DB2ENTRY 197
  - CEMT SET DB2CONN 432
  - CEMT SET DB2ENTRY 438
- TX
  - CEMT INQUIRE DB2CONN 187
  - CEMT INQUIRE DB2ENTRY 195
  - CEMT SET DB2CONN 428
  - CEMT SET DB2ENTRY 436
- TXID
  - CEMT INQUIRE DB2CONN 187
  - CEMT INQUIRE DB2ENTRY 194
  - CEMT SET DB2CONN 427
  - CEMT SET DB2ENTRY 435

TYPE  
 CEMT INQUIRE DOCTEMPLATE 210  
 CEMT INQUIRE ENQ 228  
 CEMT INQUIRE JMODEL 252  
 CEMT INQUIRE JOURNALNAME 255  
 CEMT INQUIRE REQUESTMODEL 293  
 CEMT INQUIRE TDQUEUE 330  
 CEMT INQUIRE UOWENQ 369  
 CEMT INQUIRE UOWLINK 373

TYPETERM  
 CEDA 112

**U**

U  
 CEMT INQUIRE TASK 316

UDATAKEY  
 CEMT INQUIRE TRANSACTION 342

UDSASIZE  
 CEMT INQUIRE DSAS 213

UEXECKEY  
 CEMT INQUIRE PROGRAM 289

UKOPEN  
 CEMT INQUIRE TASK 316

UNATTEMPTED  
 CEMT INQUIRE CONNECTION 177

UNAUTH  
 CEMT INQUIRE CORBASERVER 185

UNAVAILABLE  
 CEMT INQUIRE DSNAME 218  
 CEMT INQUIRE UOWLINK 373  
 CEMT INQUIRE VTAM 381  
 CEMT SET DSNAME 449  
 INQUIRE TCPIP SERVICE 323

UNBALANCD PARENS AT XXXXX  
 CMSG 586

UNCONNECTED  
 CEMT INQUIRE TSPool 351  
 CEMT INQUIRE UOWLINK 373

UNDEFINED  
 CEMT INQUIRE UOWDSNFAIL 363

UNDETERMINED  
 CEMT INQUIRE DSNAME 221

UNENABLED  
 CEMT INQUIRE FILE 239

unit of work resynchronization transaction, CPIA 589

UNKNOWN  
 CEMT INQUIRE UOWLINK 373

UNQUIESCED  
 CEMT INQUIRE DSNAME 220  
 CEMT SET DSNAME 450

UNREGISTERED  
 CEMT INQUIRE VTAM 381  
 INQUIRE TCPIP SERVICE 323

UNSUP OPRT TRMNL XXXXXXXX  
 CMSG 586

UNSUP TERMINAL XXXXXXXX  
 CMSG 586

UOW  
 CEMT INQUIRE DB2CONN 186  
 CEMT INQUIRE DB2ENTRY 194

UOW (continued)  
 CEMT INQUIRE ENQ 229  
 CEMT INQUIRE TASK 316  
 CEMT INQUIRE UOW 359  
 CEMT INQUIRE UOWDSNFAIL 365  
 CEMT INQUIRE UOWENQ 370  
 CEMT INQUIRE UOWLINK 374  
 CEMT SET DB2CONN 427  
 CEMT SET transaction 511  
 CEMT transaction 356

UOWACTION  
 CEMT INQUIRE CONNECTION 180  
 CEMT INQUIRE DSNAME 221

UOWDSNFAIL  
 CEMT INQUIRE transaction 361  
 CEMT INQUIRE UOWDSNFAIL 365

UOWENQ  
 CEMT transaction 366

UOWLINK  
 CEMT INQUIRE transaction 371  
 CEMT INQUIRE UOWLINK 374  
 CEMT SET transaction 512

UOWSTATE  
 CEMT INQUIRE UOW 359

UPDATE  
 CEMT INQUIRE FILE 241  
 CEMT SET FILE 458

UPDATEMODEL  
 CEMT INQUIRE FILE 241

UPDATESTATUS  
 CEMT INQUIRE FILE 241

uppercase input to transactions 4

URID  
 CEMT INQUIRE EXCI 233  
 CEMT INQUIRE UOWLINK 374

URIMAP  
 CEDA 112  
 CEMT DISCARD 150  
 CEMT INQUIRE transaction 375  
 CEMT INQUIRE WEBSERVICE 385  
 CEMT PERFORM STATISTICS 410  
 CEMT SET transaction 513

URM  
 CEMT SET TCPIP SERVICE 499

USAGE  
 CEMT INQUIRE URIMAP 376

USECOUNT  
 CEMT INQUIRE PROGRAM 290  
 CEMT INQUIRE STREAMNAME 300

user storage, checking 607

USERID 530  
 CEMT INQUIRE BR FACILITY 166  
 CEMT INQUIRE DB2CONN 188  
 CEMT INQUIRE DB2ENTRY 195  
 CEMT INQUIRE NETNAME 274  
 CEMT INQUIRE TASK 316  
 CEMT INQUIRE TDQUEUE 330  
 CEMT INQUIRE TERMINAL 337  
 CEMT INQUIRE UOW 360  
 CEMT INQUIRE URIMAP 378  
 CEMT SET DB2CONN 428

## USERID (continued)

CEMT SET DB2ENTRY 436

## USERTABLE

CEMT INQUIRE FILE 241

CEMT SET FILE 458

## V

### VALID

CEMT INQUIRE DSNAME 222

### VALIDATIONST

CEMT INQUIRE WEBSERVICE 386, 518

### VALIDITY

CEMT INQUIRE DSNAME 222

### value

CEKL SET TASK 132

CEMT INQUIRE BEAN 162

CEMT INQUIRE BRFACILITY 165

CEMT INQUIRE CONNECTION 173

CEMT INQUIRE CORBASERVER 183

CEMT INQUIRE DJAR 206

CEMT INQUIRE DSNAME 214

CEMT INQUIRE FILE 235

CEMT INQUIRE HOST 244

CEMT INQUIRE MODENAME 263

CEMT INQUIRE NETNAME 269

CEMT INQUIRE PARTNER 275

CEMT INQUIRE PROFILE 282

CEMT INQUIRE PROGRAM 283

CEMT INQUIRE REQUESTMODEL 291

CEMT INQUIRE STREAMNAME 299

CEMT INQUIRE SYDUMPCODE 302

CEMT INQUIRE TASK 311

CEMT INQUIRE TCLASS 317

CEMT INQUIRE TDQUEUE 326

CEMT INQUIRE TERMINAL 331

CEMT INQUIRE TRANSACTION 338

CEMT INQUIRE TRDUMPCODE 344

CEMT INQUIRE TSMODEL 347

CEMT INQUIRE TSPPOOL 350

CEMT INQUIRE TSQUEUE 352

CEMT INQUIRE UOW 356

CEMT INQUIRE UOWLINK 371

CEMT INQUIRE URIMAP 375

CEMT INQUIRE WORKREQUEST 388

CEMT SET BRFACILITY 416

CEMT SET CONNECTION 418

CEMT SET CORBASERVER 424

CEMT SET DSNAME 447

CEMT SET ENQMODEL 452

CEMT SET FILE 454

CEMT SET JOURNALNAME 463

CEMT SET LINE 466

CEMT SET MODENAME 468

CEMT SET NETNAME 473

CEMT SET PIPELINE 476

CEMT SET PROGRAM 479

CEMT SET SYDUMPCODE 487

CEMT SET TASK 493

CEMT SET TCLASS 495

CEMT SET TDQUEUE 501

## value (continued)

CEMT SET TERMINAL 503

CEMT SET TRANSACTION 506

CEMT SET TRDUMPCODE 508

CEMT SET TSQUEUE 510

CEMT SET UOW 511

CEMT SET UOWLINK 512

CEMT SET WEBSERVICE 518

CEMT SET WORKREQUEST 519

### value option

CEMT INQUIRE REQUESTMODEL 292

### values

hexadecimal 143

### View Compiled Debugging Profile screen

application debugging profile manager

3270 interface 50

### View compiled profile page

application debugging profile manager

web interface 31

### View Java Debugging Profile screen

application debugging profile manager

3270 interface 51

### View Java profile page

application debugging profile manager

web interface 32

### viewing debugging profiles

with the 3270 interface 40

with the web interface 22

### VSAM

CEMT INQUIRE DSNAME 216

CEMT INQUIRE FILE 237

### VTAM

CEMT INQUIRE CONNECTION 175

CEMT INQUIRE transaction 380

CEMT INQUIRE VTAM 381

CEMT PERFORM STATISTICS 410

CEMT SET transaction 515

VTAM ACB (access control block) 380, 515

VTAM terminals, acquiring 473, 503

## W

### WAIT

CEMT INQUIRE TASK 315

CEMT INQUIRE TDQUEUE 329

CEMT INQUIRE TRANSACTION 341

### WAITCAUSE

CEMT INQUIRE UOW 360

### WAITCOMMIT

CEMT INQUIRE UOW 360

### WAITER

CEMT INQUIRE ENQ 228

CEMT INQUIRE UOWENQ 369

### WAITFORGET

CEMT INQUIRE UOW 360

### WAITING

CEMT INQUIRE ENQMODEL 231

CEMT INQUIRE UOW 360

### WAITRMI

CEMT INQUIRE UOW 360

WAITSTATE  
 CEMENT INQUIRE UOW 360  
web interface  
 application debugging profile manager 16  
Web support 517  
 CEMENT INQUIRE transaction 382  
 CEMENT SET transaction 517  
Web support inquiry 382  
WEBSERVICE  
 CEDA 112  
 CEMENT DISCARD 151  
 CEMENT INQUIRE transaction 383  
 CEMENT INQUIRE URIMAP 378  
 CEMENT PERFORM STATISTICS 410  
 CEMENT SET WEBSERVICE 518  
work requests  
 CEMENT INQUIRE requests 387  
 CEMENT SET requests 519  
WORKREQUEST  
 CEMENT SET transaction 519  
WORKREQUEST command  
 CEMENT INQUIRE transaction 387  
WORKREQUEST option  
 INQUIRE WORKREQUEST command 390  
WORKTYPE option  
 INQUIRE WORKREQUEST command 390  
write to console operator (CWTO) 619  
WSBIND  
 CEMENT INQUIRE WEBSERVICE 385  
WSDIR  
 CEMENT INQUIRE PIPELINE 279  
 CEMENT INQUIRE WEBSERVICE 386  
WSDLFILE  
 CEMENT INQUIRE WEBSERVICE 385

## X

XCF  
 CEMENT INQUIRE CONNECTION 175  
XLNSTATUS  
 CEMENT INQUIRE CONNECTION 180  
XLT  
 CEMENT PERFORM SHUTDOWN 405  
XM  
 CEMENT INQUIRE CONNECTION 175  
XNOTDONE  
 CEMENT INQUIRE CONNECTION 180  
XOK  
 CEMENT INQUIRE CONNECTION 180  
XRF and the CEBT transaction 79  
XRF and the CEKL transaction 127

## Y

YES  
 CMSG 571, 572  
YY.DDD  
 CMSG 568  
YYYY  
 CSPG 613

YYYY.DDD  
 CMSG 570

## Z

ZERO LENGTH MSG AT "  
 CMSG 586

---

## Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply in the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

---

## Trademarks

IBM, the IBM logo, and [ibm.com](http://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](http://www.ibm.com/legal/copytrade.shtml).

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

Other product and service names might be trademarks of IBM or other companies.

---

## Sending your comments to IBM

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book.

Please limit your comments to the information in this book and the way in which the information is presented.

To ask questions, make comments about the functions of IBM products or systems, or to request additional publications, contact your IBM representative or your IBM authorized remarketer.

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

You can send your comments to IBM in any of the following ways:

- By mail, to this address:

IBM United Kingdom Limited  
User Technologies Department (MP095)  
Hursley Park  
Winchester  
Hampshire  
SO21 2JN  
United Kingdom

- By fax:
  - From outside the U.K., after your international access code use 44-1962-816151
  - From within the U.K., use 01962-816151
- Electronically, use the appropriate network ID:
  - IBMLink: HURSLEY(IDRCF)
  - Internet: idrcf@hursley.ibm.com

Whichever you use, ensure that you include:

- The publication title and order number
- The topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.







Product Number: 5655-M15

SC34-6432-07



Spine information:



CICS TS for z/OS

CICS Supplied Transactions

Version 3  
Release 1