CICS® Transaction Server for OS/390®

IBM

CICS Supplied Transactions

Release 3

CICS® Transaction Server for OS/390®

CICS Supplied Transactions

Release 3

┌─ **Note!** ───┐

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

└──┘

Fourth edition (November 2000)

Contents

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

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

ACF/VTAM	IBM
BookManager	IBMLink
CICS	IMS
CICS/ESA	IMS/ESA
CICS/VM	Language Environment
CICSPlex	MQ
DB2	MVS/DFP
DFSMS	OS/390
DFSMS/MVS	RACF
DFSMSdss	VTAM
DFSMShsm	

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

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

Preface

What this book is about

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

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

Who should read this book

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

What you need to know to understand this book

A general knowledge of the concepts and terminology is required.

How to use this book

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

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

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

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

Front end programming interface

Information about the Front End Programming Interface 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 OS/390 Release 3
- **VTAM**® refers to IBM ACF/VTAM®
- **IMS**™ refers to IBM IMS/ESA®
- **TCAM** refers to the DCB interface of ACF/TCAM.

Bibliography

CICS Transaction Server for OS/390

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

CICS books for CICS Transaction Server for OS/390

General

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

Administration

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

Programming

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

Diagnosis

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

Communication

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

Special topics

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

CICSPlex SM books for CICS Transaction Server for OS/390

General

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

Administration and Management

CICSPlex SM Administration	SC34-5401
CICSPlex SM Operations Views Reference	SC33-0789
CICSPlex SM Monitor Views Reference	SC34-5402
CICSPlex SM Managing Workloads	SC33-1807
CICSPlex SM Managing Resource Usage	SC33-1808
CICSPlex SM Managing Business Applications	SC33-1809

Programming

CICSPlex SM Application Programming Guide	SC34-5457
CICSPlex SM Application Programming Reference	SC34-5458

Diagnosis

CICSPlex SM Resource Tables Reference	SC33-1220
CICSPlex SM Messages and Codes	GC33-0790
CICSPlex SM Problem Determination	GC33-0791

Other CICS books

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

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

Books from related libraries

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

Determining if a publication is current

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

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

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

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

Summary of changes

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

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

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 Chapter 2, "CBAM—BTS browser" on page 11).

- 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 release 2 edition

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 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 OS/390, 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.

Changes for the CICS/ESA 4.1 edition

This edition is based on the CICS/ESA 3.3 *CICS-Supplied Transactions* manual (SC33-0669-02).

For CICS/ESA 4.1, the following changes have been made:

- CEMT INQUIRE CONNECTION has been enhanced to display more information.

- CEMT INQUIRE|SET DSAS has been added.

- CEMT INQUIRE|PERFORM|SET DELETSHIPPED has been added.

- RELATED and LOCAL options have been added to the CEMT INQUIRE SYDUMPCODE and TRDUMPCODE commands and to the CEMT SET SYDUMPCODE and TRDUMPCODE commands.

- CEMT, CEDA, and CETR now show the SYSID alongside the APPLID.

- CEDA now shows the resource type and resource name in all the View, Alter, and Define panels.

- In CEMT INQ|SET SYSTEM, the CSCS, ECSCS, ERSCS, EUSCS, and USCS operands have been removed, and the DSALIMIT, EDSALIMIT, ESDASIZE, RDSASIZE, SDSASIZE, and TRANISOLATE operands have been added. In addition, the definitions have been changed for the CDSASIZE, ECDSASIZE, ERDSASIZE, and EUDSASIZE operands. The DTRTRANSID, PROGAUTOCTLG, PROGAUTOEXIT, and PROGAUTOINST operands have also been added. FEPI and PROGAUTO options are added to the CEMT PERFORM STATISTICS command.

In addition, many editorial changes have been made to clarify the information in this manual.

Chapter 1. Transactions supplied by CICS

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 Appendix, "List of CICS transactions" on page 269.

Function	Name	Page
CICS business transaction services (BTS) browser	CBAM	11
Command-level interpreter	CECI, CECS	33
Database control inquiry	CDBI	15
Database control interface	CDBM	17
Database control menu	CDBC	13
Execution diagnostic facility	CEDF	43
	CEDX	43
In-doubt testing tool	CIND	231
Master terminal	CEMT	47
Master terminal (alternate CICS)	CEBT	23
Messages and codes display	CMAC	235
Message switching	CMSG	237
Page retrieval	CSPG	259
Remote transactions	CRTE	253
Resource definition online (RDO)	CEDA, CEDB, CEDC	35
Sign off	CESF	219
Sign on	CESN	221
Supervisory terminal	CEST	223
Temporary-storage browse	CEBR	21
Terminal status	CEOT	217
Trace control	CETR	225
Terminal and system test	CSFE	255
Write to console operator	CWTO	265

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

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

Upgrading CICS transactions

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.

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 in this book

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─┬─▶◀ 　　├─B─┤ 　　└─C─┘	A set of alternatives—one of which you **must** code.
▶▶─┬─A─┬─▶◀ 　　├─B─┤ 　　└─C─┘	A set of alternatives—one of which you **must** code. You **may** code more than one of them, in any sequence.
▶▶─┬───┬─▶◀ 　　├─A─┤ 　　├─B─┤ 　　└─C─┘	A set of alternatives—one of which you **may** code.
▶▶─┬───┬─▶◀ 　　├─A─┤ 　　├─B─┤ 　　└─C─┘	A set of alternatives — any number (including none) of which you may code once, in any sequence.
▶▶─┬─A─┬─▶◀ 　　└─B─┘	Alternatives where **A** is the default.
▶▶─┤ Name ├─▶◀ **Name:** ├─A─┬─┤ 　　└─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.

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 operators

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.

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 "The system console as a CICS terminal" on page 8.

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 49 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.

The system console as a CICS terminal

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 TERMINAL command. If this has not been done, you get the following message when you try to use the console:

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

and your input is ignored.

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

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

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

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

Rules for console entry

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

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

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

MODIFY command

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

```
MODIFY ident,datastring
```

You can abbreviate the MODIFY command to F.

`ident` can be any of the following:

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

  ```
  START procedure_name
  ```

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

  ```
  START procedure_name.taskid
  ```

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

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

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

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

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

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

```
@nn DFH4200A jjjjjjjj tttt
```

where:

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

jjjjjjjj   is the jobname of CICS in the operating system.

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

REPLY command

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

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

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

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

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

Example of a conversation using CONVERSE

```
modify job002,serv   1

@17 FAULT TYPE?      2

r 17,elec     3

MESSAGE HAS BEEN SENT
```

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

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

3 Your reply that the fault is an electrical one.

Example of a conversation using WRITE/READ

```
modify job002,usid     1

USER SIGNON ID=?     2

@25 DFH4200A JOB002 USID    3

r 25,accts1     4

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

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

2 Application-program message produced by a WRITE command.

3 System message produced by a READ command.

4 Your reply.

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

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

Time sharing option (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.

Chapter 2. CBAM—BTS browser

Use CBAM to browse the CICS business transaction services (BTS) objects (process-types, processes, activities, containers, events and timers) known to this region. CBAM is described in the *CICS Business Transaction Services* manual.

Chapter 3. 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 1. 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.

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 2. 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.

13

CDBC—database control menu

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 3. 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*.

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 4. CDBC transaction: the HELP panel

Chapter 4. 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:

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




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






 PF1 = Help   2 = Refresh   3 = End

```

Figure 5. CDBI transaction: initial screen

This display tells you the current status of the CICS-DBCTL connection, which can 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.

The 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 6. CDBI transaction: the HELP panel

CDBI–database control inquiry

Chapter 5. 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.

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.

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 7.

```
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 7. 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 8.

```
CDBM                    CICS-DBCTL Operator Transaction          98.135
                                                                13:24:20
 Type IMS command.
                /GROUP SAMPLE STA_____
                _____
                _____
                _____

 For /DBDUMP or /DBRECOVER commands

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


 Press enter to display responses.

 CICS APPLID DBDCCICS
 DBCTL ID    SYS3

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

Figure 8. CICS-DBCTL operator transaction panel showing a GROUP command

17

| **Note:** The group command *SAMPLE STA* referred to in this
| panel can be found in the "DFHDBFK - The CDBM
| GROUP command file" on page 18.

| Responses to commands issued using CDBM are returned in
a screen like the one shown in Figure 9, which shows the
first of a number of screens issued in response to a
/DISPLAY DB ALL command.

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

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

Figure 9. CICS-DBCTL IMS responses

| 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
| and to maintain its content. One method of maintainting 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* manual.

| 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 *CICS IMS Database Control Guide* for details). Leading or embedded blanks are not allowed, but trailing blanks are acceptable. **Note:** The validity of the IMS command verb is not checked by CDBM. Invalid values will be reported by IMS when the command is attempted.
3	1406	IMS Command parameters	Up to 1406 characters of parameters appropriate to the chosen IMS command verb. (This will often consist of lists of databases.) **Note:** Wildcard characters may not be used in the parameters stored in the CDBM Group command file. This is unlike the other functions of the CDBM transaction which permit the use of wildcard characters to describe multiple similarly named databases.

| The MAINTENANCE panel for DFHDBFK

| If you press the Maintenance key (PF2) on the main CDBM
| panel, you get the panel shown in Figure 10 on page 19.

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

  F1=Help  F3=Exit  F12=Cancel
```
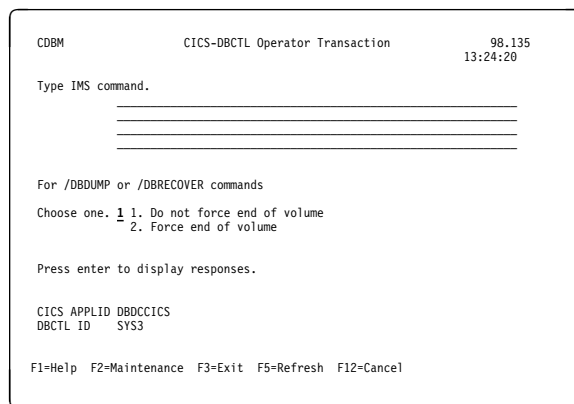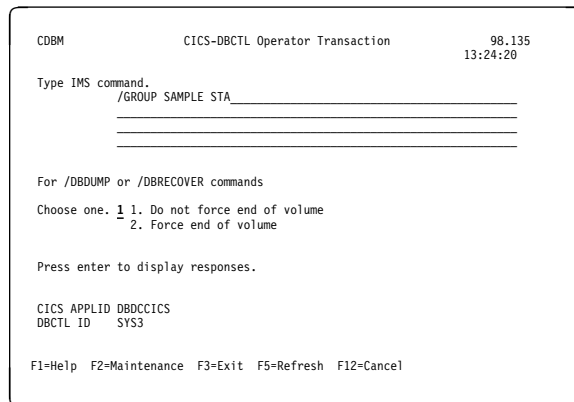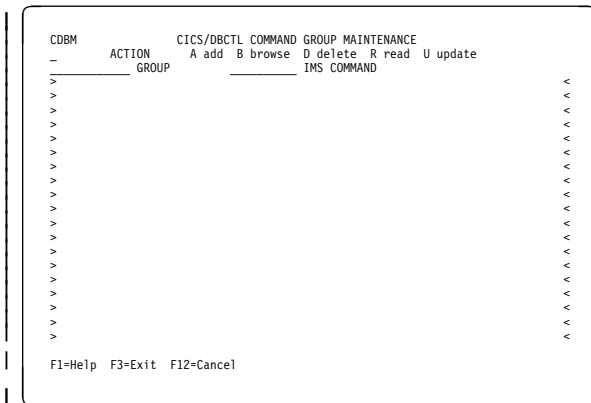
Figure 10. CICS-DBCTL Group Maintenance panel

Input fields

The input fields are:

- Action

- Group

- IMS Command

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

Group, IMS Command and IMS Command parameters are described in the section "Record layout in the CDBM GROUP command file" on page 18

The Action field will accept one of the following:

A   Add

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

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

B   Browse

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

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

D   Delete

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

R   Read

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

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

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

U   Update

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

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

Note: Use Add to create a new key.

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

The CICS-DBCTL interface

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*.

The HELP panels

If you press the help key (PF1) from the CICS-DBCTL Operator Transaction panel, you get the panel shown in Figure 11 on page 20.

CDBM–database control interface

```
  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 11. 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 12.

```
  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 12. CICS-DBCTL Maintenance help panel

Chapter 6. CEBR—temporary storage browse

Use the CEBR transaction to browse temporary storage queues and to delete them.

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

| Remember that:

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

| This is can lead to confusion if for example an EXEC
| CICS READQ NEXT and a CEBR transaction attempt to
| work with the same Temporary Storage queue at the
| same time.

| • transient data queue reads are destructive. If you read a
| transient data queue that is being used on an active
| system by applications, this is likely to cause problems.

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

```
CEBR CEBRS209
```

CICS responds with a display of the queue, a sample of which is shown in Figure 13:

```
   CEBR  TSQ AXBYQUEUENAME1   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 13. 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 35 on page 43) 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 14.

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 ===>                                                 ▮2▮
      ************************** TOP OF QUEUE ****************************
      *
      ************************** BOTTOM OF QUEUE *************************



                                                        ▮3▮




 TSQUEUE AXBYQUEUEAME1 DOES NOT EXIST                              ▮4▮
 PF1 : HELP              PF2 : SWITCH HEX/CHAR     PF3 : TERMINATE BROWSE ▮5▮
 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 14. 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*.

The 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.
```

CEBR—temporary storage browse

Chapter 7. 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 "The system console as a CICS terminal" on page 8.

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|SET AUXTRACE)
- The dump data set (CEBT INQUIRE|SET DUMP)
- GTF tracing (CEBT INQUIRE|SET GTFTRACE)
- Internal tracing (CEBT INQUIRE|SET INTTRACE)
- Surveillance or 'tracking' of the active CICS system by the alternate CICS system (CEBT INQUIRE|SET SURVEILLANCE)
- Takeover from the active CICS system by the alternate CICS system (CEBT INQUIRE|SET TAKEOVER).

Descriptions follow of the CEBT INQUIRE and SET commands in alphabetic order of the name of the resource.

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──►◄
```

Displayed fields

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

Auxiliary
identifies this panel as relating to auxiliary trace.

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

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

Noswitch
Switching does not occur without operator intervention.

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

Start
Auxiliary tracing is in progress.

Stop

Auxiliary tracing has stopped.

Switch

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

CEBT INQUIRE DUMP

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──►◄
```

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
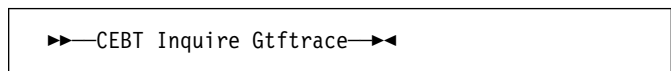
Function: Show the GTF status.

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──►◄
```

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.
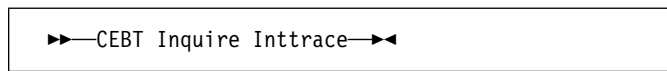
CEBT INQUIRE INTTRACE

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──►◄
```

Displayed fields

Inttrace
> indicates that this panel relates to an INTTRACE inquiry.

Start
> CICS internal tracing is in progress.

Stopped
> CICS internal tracing is not in progress.

CEBT INQUIRE SURVEILLANCE

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──►◄
```

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

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)
- Request a dump of the alternate CICS and continue processing (CEBT PERFORM SNAP)
- Cause the alternate CICS to take over from the active CICS (CEBT PERFORM TAKEOVER).

Descriptions follow of the CEBT PERFORM commands in alphabetic order.

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──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 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──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.

CEBT SET AUXTRACE

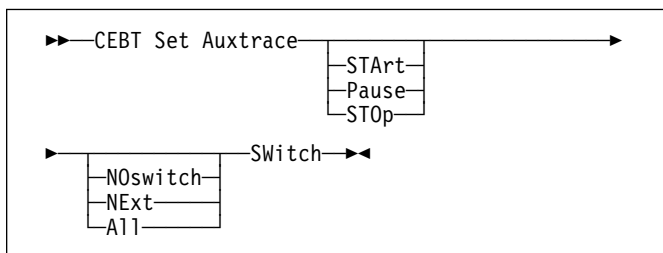**Function:** Change auxiliary tracing options.

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 49).

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────────────────────────►
                        ├─STArt─┤
                        ├─Pause─┤
                        └─STOp──┘

►───────────────────SWitch──►◄
    ├─NOswitch─┤
    ├─NExt─────┤
    └─All──────┘
```

Options

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

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

NOswitch
 Switching between the auxiliary trace data sets is not to occur without operator intervention.

Pause
 Auxiliary tracing is to stop, but the data set is to remain open. A subsequent START request causes trace entries to be written immediately following those that were written before the PAUSE request.

STArt
 CICS is to start auxiliary tracing and open the auxiliary trace data set if it is currently closed.

STOp
 CICS is to stop auxiliary tracing and close the auxiliary trace data set. A subsequent START request causes new trace entries to be written at the start of the data set, thereby overwriting the trace entries that were written before the STOP request.

SWitch
 specifies that an immediate switch from the current auxiliary trace data set is to be made to the alternate trace data set.
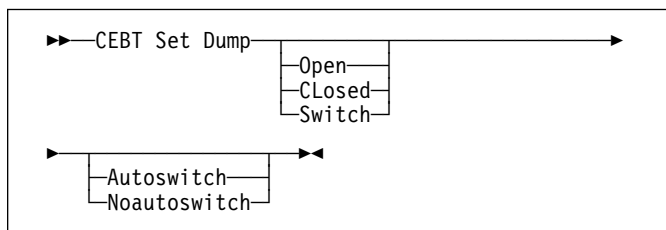
CEBT SET DUMP

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 overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see "Overtyping a display" on page 49).

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

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

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

- Type `CEBT SET DUMP` (`CEBT S DU`) followed by one or more attribute settings that you wish to change. For example, `CEBT S DU A` will cause an automatic switch to the inactive dump data set if 'noautoswitch' was previously specified.

```
►►──CEBT Set Dump────────────────────────────►
                    ├─Open───┤
                    ├─CLosed─┤
                    └─Switch─┘

►──────────────────────────►◄
    ├─Autoswitch───┤
    └─Noautoswitch─┘
```

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

**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 overtype them with the required values.

- Type `CEBT SET GTFTRACE` (`CEBT S G`) followed by one or more attribute settings that you wish to change. For example, `CEBT S G STO` will stop GTF tracing if it is already running.
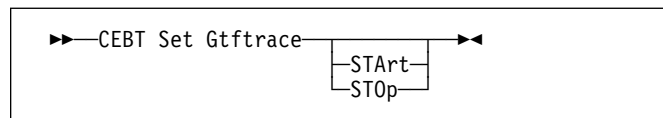
```
►►──CEBT Set Gtftrace───────────►◄
                       ├─STArt─┤
                       └─STOp──┘
```

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 INTTRACE

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

• Type CEBT SET INTTRACE (CEBT S IN) followed by one or more attribute settings that you wish to change. For example, CEBT S IN STO will stop internal tracing if it is already running.

```
►►──CEbt Set Inttrace──────────►◄
                    ├─STArt─┤
                    └─STOp──┘
```
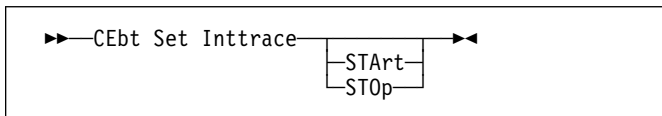
Options

STArt
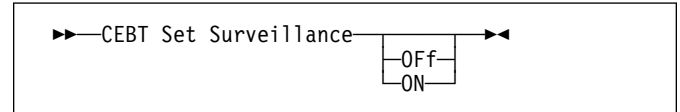   CICS internal tracing is started.

STOp
    CICS internal tracing is stopped.

CEBT SET SURVEILLANCE

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

Syntax

```
►►──CEBT Set Surveillance───────►◄
                       ├─OFf─┤
                       └─ON──┘
```

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 overtype them with the required values.

- Type CEBT SET TAKEOVER (CEBT S T) followed by one or more attribute settings that you wish to change. For example, CEBT S T M changes takeover to a manual operation if it was previously specified to be automatic.

```
►►──CEBT Set Takeover──────────────►◄
                    ─Automatic─
                    ─Command──
                    ─Manual───
```
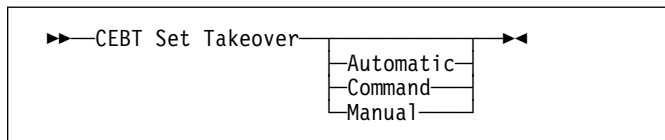
Options

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.

The 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 modify results, position cursor using TAB key, overtype and press ENTER.
? at start of results line gives SYNTAX OF SET COMMAND.

PF KEY HELP INFORMATION

  PF01  HELP  HELP INFORMATION
  PF02        UNDEFINED
  PF03  END   END SESSION
  PF04        UNDEFINED
  PF05        UNDEFINED
  PF06        UNDEFINED
  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
```
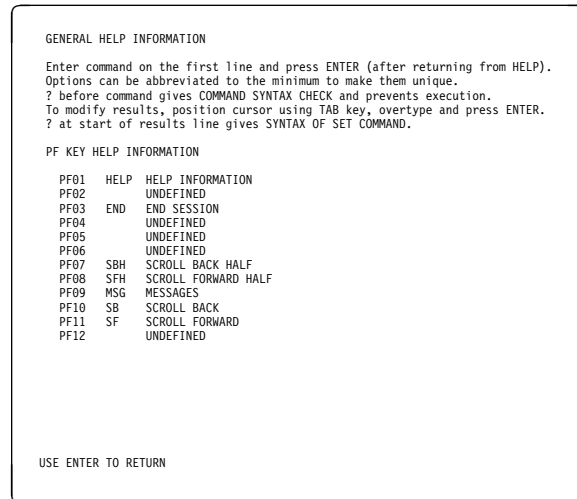
Figure 15. CEBT transaction: the HELP panel

Chapter 8. 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* manual or the *CICS System Programming Reference* manual.

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       POInt       ROute       WAITCics
 BUild       EXtract     POP         SENd        WRITE
 CAncel      FEpi        POSt        SET         WRITEQ
 CHange      FOrmattime  PURge       SIGNOFf     Xctl
 COLlect     FREE        PUSh        SIGNON
 CONNect     FREEMain    Query       SPOOLClose
 CONVerse    GDs         READ        SPOOLOpen
 DELAy       GEtmain     READNext    SPOOLRead
 DELETE      Handle      READPrev    SPOOLWrite
 DELETEQ     IGnore      READQ       START
 DEQ         INquire     RECeive     STARTBr


 PF 1 HELP 2 HEX 3 END 4 EIB 5 VAR 6 USER          9 MSG
```

Figure 16. CECI transaction: initial screen

Note: RECEIVE and SEND cannot be used on TCAM-attached terminals. 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 17 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 17. 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* manual for programming information, including a "translation" of these numeric values to their alphabetic values.

The 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 18. 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 16 on page 33. 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 9. 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 19), 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 19. CEDA transaction: initial screen

CEDA provides the following functions:

ADd   Adds a group name to a list, and creates a new list if the list did not already exist.

ALter   Changes some of the attributes of an existing resource definition.

APpend   Copies a list to another list.

CHeck   Checks that the resource definitions within a group or list are consistent with one another.

COpy   Copies one or more resource definitions.

DEFine   Creates a new resource definition on the CSD file.

DELete   Erases one or more resource definitions from the CSD file.

DIsplay   Displays one or more lists or groups on a full-screen panel.

Expand   Shows the resource definitions in one or more groups on a full-screen panel.

Install   Dynamically makes one or more resource definitions available to an active CICS system.

Lock   Assures exclusive access to a group or list by restricting update and delete access to a single operator identifier.

Move   Moves one or more resource definitions from one named group to another.

REMove   Removes a group name from a list.

REName   Renames a resource definition.

UNlock   Removes a lock on a group or list which was previously put there by a CEDA LOCK command.

USerdefine 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.

View   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 20 which lists the resources:

```
  DEF
   ENTER ONE OF THE FOLLOWING

  Connection   TCpipservice
  DB2Conn      TDqueue
  DB2Entry     TErminal
  DB2Tran      TRANClass
  DOctemplate  TRANSaction
  Enqmodel     TSmodel
  File         TYpeterm
  Journalmodel
  Lsrpool
  Mapset
  PARTItionset
  PARTNer
  PROCesstype
  PROFile
  PROGram
  Requestmodel
  Sessions

                              SYSID=JOHN APPLID=I

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

Figure 20. CEDA transaction: DEFINE screen

CEDA—resource definition online

If you enter INST on the command line and press ENTER, you get the screen shown in Figure 21 which lists the resources:
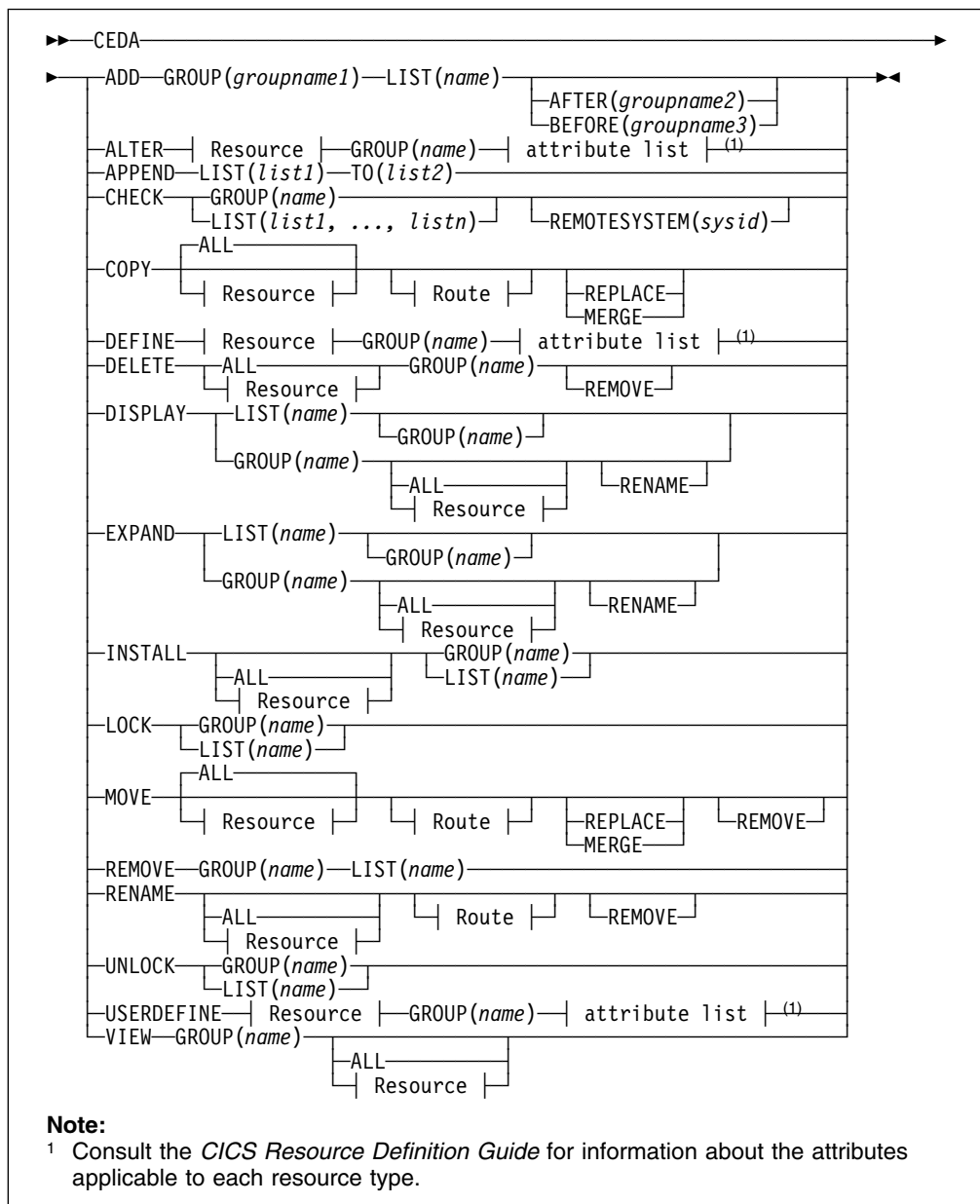
```
 CEDA INST
 OVERTYPE TO MODIFY
  CEDA  Install
   All
   Connection   ==>
   DB2Conn      ==>
   DB2Entry     ==>
   DB2Tran      ==>
   DOctemplate  ==>
   Enqmodel     ==>
   File         ==>
   Journalmodel ==>
   LSrpool      ==>
   Mapset       ==>
   PARTItionset ==>
   PARTNer      ==>
   PROCesstype  ==>
   PROFile      ==>
   PROGram      ==>
   Requestmodel ==>
   Sessions     ==>
   TCpipservice ==>
   TDqueue      ==>
   TErminal     ==>
 + TRANClass    ==>
 + TRANSaction  ==>
   TSmodel      ==>
   TYpterm      ==>
   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 21. CEDA transaction: INSTALL screen

Command syntax

```
►►──CEDA────────────────────────────────────────────────────────────────►

►──┬─ADD──GROUP(groupname1)──LIST(name)──────────────────────────────┬──►◄
   │                           ┌─AFTER(groupname2)──┐                 │
   │                           └─BEFORE(groupname3)─┘                 │
   ├─ALTER──┤ Resource ├──┬─GROUP(name)─┬── attribute list ├─(1)──────┤
   ├─APPEND──LIST(list1)──TO(list2)─────────────────────────────────┤
   ├─CHECK──┬─GROUP(name)──────────────┬──┬─REMOTESYSTEM(sysid)─┬────┤
   │        └─LIST(list1, ..., listn)──┘                         │    │
   │        ┌─ALL──────────────┐                                 │    │
   ├─COPY───┼──────────────────┤──┤ Route ├──┬─REPLACE─┬─────────┤
   │        └─┤ Resource ├──────┘              └─MERGE───┘         │
   ├─DEFINE──┤ Resource ├──GROUP(name)── attribute list ├─(1)──────┤
   ├─DELETE──┬─ALL────────────┬──GROUP(name)──┬─REMOVE─┬──────────┤
   │         └─┤ Resource ├────┘                └───────┘          │
   ├─DISPLAY──┬─LIST(name)────────────────────────────────┬───────┤
   │          │              ┌─GROUP(name)─┐               │       │
   │          └─GROUP(name)──┴─────────────┴──┬─ALL──────┬─┬─RENAME─┤
   │                                           └─┤Resource├┘        │
   ├─EXPAND──┬─LIST(name)─────────────────────────────────┬────────┤
   │         │             ┌─GROUP(name)─┐                │        │
   │         └─GROUP(name)──┴─────────────┴──┬─ALL──────┬──┬─RENAME─┤
   │                                          └─┤Resource├┘         │
   ├─INSTALL──┬─ALL────────────┬──┬─GROUP(name)─┬──────────────────┤
   │          └─┤ Resource ├────┘  └─LIST(name)──┘                  │
   ├─LOCK──┬─GROUP(name)─┬──────────────────────────────────────────┤
   │       └─LIST(name)──┘                                          │
   │        ┌─ALL──────────────┐                                    │
   ├─MOVE───┼──────────────────┤──┤ Route ├──┬─REPLACE─┬──┬─REMOVE─┤
   │        └─┤ Resource ├──────┘              └─MERGE───┘  └───────┘
   ├─REMOVE──GROUP(name)──LIST(name)────────────────────────────────┤
   ├─RENAME──┬─ALL────────────┬──┤ Route ├──┬─REMOVE─┬──────────────┤
   │         └─┤ Resource ├────┘              └───────┘             │
   ├─UNLOCK──┬─GROUP(name)─┬─────────────────────────────────────────┤
   │         └─LIST(name)──┘                                        │
   ├─USERDEFINE──┤ Resource ├──GROUP(name)── attribute list ├─(1)────┤
   └─VIEW──GROUP(name)──┬─ALL──────────┬─────────────────────────────┘
                        └─┤ Resource ├─┘
```

Note:

1. Consult the *CICS Resource Definition Guide* for information about the attributes applicable to each resource type.

CEDA—resource definition online

```
Resource:
├──────CONNECTION(name)─────────┤
│      ─DB2CONN(name)────────│
│      ─DB2ENTRY(name)───────
│      ─DB2TRAN(name)────────
│      ─DOCTEMPLATE(name)────
│      ─ENQMODEL(name)───────
│      ─FILE(name)───────────
│      ─JOURNALMODEL(name)───
│      ─LSRPOOL(name)────────
│      ─MAPSET(name)─────────
│      ─PARTITIONSET(name)───
│      ─PARTNER(name)────────
│      ─PROCESSTYPE(name)────
│      ─PROFILE(name)────────
│      ─PROGRAM(name)────────
│      ─REQUESTMODEL(name)───
│      ─SESSIONS(name)───────
│      ─TCPIPSERVICE(name)───
│      ─TDQUEUE(name)────────
│      ─TERMINAL(name)───────
│      ─TRANCLASS(name)──────
│      ─TRANSACTION(name)────
│      ─TSMODEL(name)────────
│      ─TYPETERM(name)───────

Route (Copy and Move):
├──GROUP(from-name)──┬─AS(newname)───────────┬─┤
                     ─TO(to-name)─────────
                     ─AS(newname)──TO(to-name)─

Route (Rename):
├──────┬─────────────────┬──┬──────────────┬──┬─────────────┬──┤
       ─GROUP(from-name)─    ─AS(newname)─    ─TO(to-name)─

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.

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.

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.

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.

CSD compatibility panel

Figure 22 on page 40 shows a normal ALTER panel with one attribute (PRIMedsize) that is obsolete in CICS/ESA® 3.3, protected as indicated by the colon symbol (:).

Use the PF2 key to change to the compatibility mode panel shown in Figure 25 on page 40. PRIMedsize is now shown with the ==> symbols indicating that it can be modified. Use the PF2 key again to return to the normal panel.

See the *CICS Resource Definition Guide* for background information about the CSD compatibility panel.

```
ALTER TRANS
OVERTYPE TO MODIFY                              CICS RELEASE = 0410
 CEDA  ALter TRANSaction(    )
  TRANSaction ==>
  Group       ==>
  DEscription ==>
  PROGram     ==>
  TWasize     ==>            0-32767
  PROFile     ==>
  PArtitionset ==>
  STAtus      ==>            Enabled | Disabled
  PRIMedsize  ==>            0-65520
  TASKDATALoc ==>            Below | Any
  TASKDATAKey ==>            User | Cics
  STOrageclear ==>           No | Yes
  RUnaway     ==>            System | 0-2700000
  SHutdown    ==>            Disabled | Enabled
  ISolate     ==>            Yes | No
 REMOTE ATTRIBUTES
+ DYnamic     ==>            No | Yes
```

Figure 22. CEDA transaction: normal panel

```
  REMOTESystem ==>
  REMOTEName   ==>
  TRProf       ==>
  Localq       ==>           No | Yes
 SCHEDULING
  PRIOrity     ==>           0-255
  TClass       :             No | 1-10
  TRANClass    ==>
 ALIASES
  Alias        ==>
  TASKReq      ==>
  XTRanid      ==>
  TPName       ==>
               ==>
  XTPname      ==>
               ==>
               ==>
 RECOVERY
  DTimout      ==>           No | 1-6800
+ INdoubt      ==>           Backout | Commit | Wait
```

Figure 23. CEDA transaction: normal panel (continued)

```
  RESTart      ==>           No | Yes
  SPurge       ==>           No | Yes
  TPUrge       ==>           No | Yes
  DUmp         ==>           Yes | No
  TRACe        ==>           Yes | No
 SECURITY
  RESSec       ==>           No | Yes
  Cmdsec       ==>           No | Yes
  Extsec       :
  TRANSec      :             1-64
  RSl          :             0-24 | Public
```

Figure 24. CEDA transaction: normal panel (end)

```
OVERTYPE TO MODIFY                     COMPATIBILITY MODE
 CEDA  ALter
  TRansaction ==>
  Group       ==>
  DEscription ==>
  PROGram     ==>
  TWasize     ==>            0-32767
  PROFile     ==>
  PArtitionset ==>
  STatus      ==>            Enabled | Disabled
  PRIMedsize  ==>            0-65520
  TASKDATALoc ==>            Below | Any
  TASKDATAKey ==>            User | Cics
 REMOTE ATTRIBUTES
  DYnamic     ==>            No | Yes
  REMOTESystem ==>
  REMOTEName  ==>
  TRProf      ==>
+ Localq      ==>            No | Yes
```

Figure 25. CEDA transaction: compatibility mode panel

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*
manual.

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 26. 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 27. 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 28. CEDA transaction: HELP Panel 2

```
                            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 31. CEDA transaction: HELP Panel 5

```
                      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 29. CEDA transaction: HELP Panel 3

```
                    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 32. CEDA transaction: HELP Panel 6

```
                       Expand and Display

The commands you can use on each kind of panel are as follows:

   EXPAND GROUP panel  - Alter, Copy, Delete, Install, Move, Rename, View
   EXPAND LIST panel   - Add, Remove
   DISPLAY GROUP panel - Check, Expand, Install , Lock, Unlock
   DISPLAY LIST panel  - Append, Check, Expand, Lock, Unlock

All these commands operate on the thing beside which you enter them.
ALTER means ALTER PROGRAM(P) GROUP(G), if these are the object and
group values on the line. In this case, since no attributes are changed,
you will see a display of the object which you can then overtype.
The EXPAND command on a DISPLAY panel also results in a new panel.

You can enter as many commands as you like at one time and you can
use = to mean the same command as the last one.

The RENAME option of EXPAND GROUP gives a panel on which you can change the
names of objects directly, by overtyping the name fields displayed.
When you change a name field a RENAME command is put in the corresponding
command field, and causes anything entered there to be ignored.

  Press Enter or any PF key to return to Help Selection Panel
```

Figure 30. CEDA transaction: HELP Panel 4

```
                            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 33. 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
 DIsplay
 Expand
 Lock
 Move
 REMove
 REName
 UNlock
 USerdefine
 View

                                       SYSID=CI41 APPLID=IYAHZCCV

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

Figure 34. 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.

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

Chapter 10. 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.

CEDF transaction

CEDF allows you to intercept your application programs at the program's initiation, at each CICS command, and at the program termination. CEDF helps you to isolate and focus on problems in your application programs.

For information on how to use the CEDF transaction, see the *CICS Application Programming Guide*.

You must ensure that the EDF resource definitions are installed. These resource definitions are provided in the IBM-supplied group, DFHEDF. For information about installing the resource definitions, see the *CICS Resource Definition Guide*.

You must also ensure that the terminal from which you run EDF is in transceive (ATI/TTI) status.

Command syntax

```
►►──CEDF──┬──────────┬──┬─,ON──┬──►◄
          ├─termid───┤  └─,OFF─┘
          ├─sysid────┤
          └─sessionid┘
```
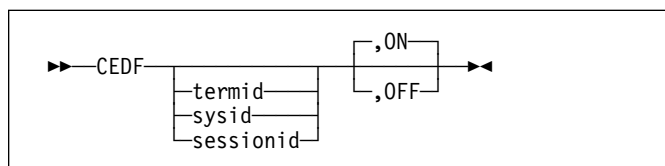
Command options

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:
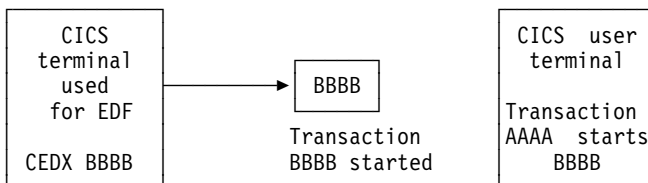
```
CEDF T33A
CEDF ,ON
```

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 35 shows the initial screen for a typical transaction using EDF.

```
 TRANSACTION: ACCT  PROGRAM: ACCT00  TASK: 0000054  APPLID: 00001349  DISPLAY: 00
 STATUS:  PROGRAM INITIATION

     EIBTIME    = 110343
     EIBDATE    = 89174
     EIBTRNID   = 'ACCT'
     EIBTASKN   = 54
     EIBTRMID   = '025L'

     EIBCPOSN   = 4
     EIBCALEN   = 0
     EIBAID     = X'7D'                          AT X'00543F1E'
     EIBFN      = X'0000'                        AT X'00543F1F'
     EIBRCODE   = X'000000000000'                AT X'00543F21'
     EIBDS      = '........'
 +   EIBREQID   = '........'

 ENTER:  CONTINUE
 PF1 : UNDEFINED          PF2 : SWITCH HEX/CHAR      PF3 : END EDF SESSION
 PF4 : SUPPRESS DISPLAYS  PF5 : WORKING STORAGE      PF6 : USER DISPLAY
 PF7 : SCROLL BACK        PF8 : SCROLL FORWARD       PF9 : STOP CONDITIONS
 PF10: PREVIOUS DISPLAY   PF11: UNDEFINED            PF12: UNDEFINED
```

Figure 35. 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 36 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 36. 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 6, "CEBR—temporary storage browse" on page 21; 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 8, "CECI—command-level interpreter" on page 33; for detailed guidance on using them, see the *CICS Application Programming Guide*.)

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

```
►►──CEDX──tranid─┬──,ON──┬──►◄
                 └──,OFF─┘
```
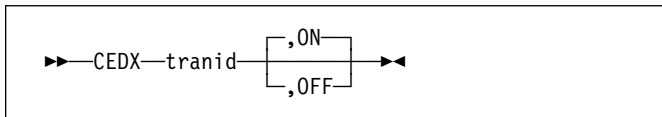
Command options

OFF

specifies that the EDF screen is to be switched OFF. If
you specify OFF you must enter the preceding comma,
as shown in the following example:

```
CEDX TRNA,OFF
```

ON

specifies that the EDF screen is to be switched ON. The
default is ON. If you specify ON you must enter the
preceding comma, as shown in the following example:

```
CEDX TRNB,ON
```

tranid

specifies the identifier of a transaction that you want to
test using the execution diagnostic facility. The
diagnostic panels are displayed on the terminal at which
you issue the CEDX *tranid* command.

The transaction you specify on the CEDX command
must run in the local CICS region(CRTE followed by
CEDX is not allowed) .

CEDX cannot interrupt transactions that are already
running when you issue the CEDX command. It affects
only a transaction that starts *after* you issue the
command.

The following diagram illustrates the use of CEDX to invoke
EDF for a transaction initiated by an EXEC CICS START
command:

```
┌────────────┐                    ┌────────────┐
│   CICS     │                    │ CICS  user │
│  terminal  │        ┌──────┐    │  terminal  │
│   used     │───────►│ BBBB │    │            │
│  for EDF   │        └──────┘    │ Transaction│
│            │                    │ AAAA  starts│
│ CEDX BBBB  │        Transaction │    BBBB    │
└────────────┘        BBBB started└────────────┘
```

The CEDX command is   AAAA issues
issued at this terminal,   EXEC CICS START
before BBBB is   command to start
started, to invoke EDF   BBBB, without
services to intercept and   a terminal.
debug the specified
transaction.

CEDF—execution diagnostic facility

Chapter 11. 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 23)
CEOT (see page 217)
CEST (see page 223).

The CECI transaction (see page 33)) 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 "The system console as a CICS terminal" on page 8. 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 "CEMT INQUIRE SYSTEM" on page 118.

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 37. 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 PROGram
 Connection   Requestmodel
 DB2Conn      TCLass
 DB2Entry     TCPipservice
 DB2Tran      TDqueue
 DOctemplate  TErminal
 Enqmodel     TRansaction
 FENode       TSmodel
 FEPOol
 FEPRopset
 FETarget
 FIle
 JModel
 JOurnalname
 PArtner
 PROCesstype
 PROFile
   S At least 4 characters of DISCARD must be specified.
                                        SYSID=JOHN APPLID=I

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

Figure 38. CEMT transaction: DISCARD screen

If you type:

I

and press ENTER, you get the following screen:

CEMT—master terminal

```
   I
   STATUS: ENTER ONE OF THE FOLLOWING OR HIT ENTER FOR DEFAULT

AUTInstmodel FENode      PROGram     TSQueue
AUTOinstall  FEPOol      REquestmodel UOW
AUXtrace     FEPRopset   RRms        UOWDsnfail
CFdtpool     FETarget    STAtistics  UOWLink
COnnection   FIle        STReamname  Vtam
DB2Conn      Gtftrace    SYDumpcode  Web
DB2Entry     INttrace    SYStem
DB2Tran      IRc         TAsk
DEletshipped JMOdel      TCLass
DOctemplate  JOurnalname TCPIP
DSAs         Line        TCPIPService
DSName       MODename    TDqueue
DUmpds       MONitor     TErminal
ENQ          Netname     TRAnsaction
ENQMOdel     PArtner     TRDumpcode
EXci         PROCesstype TSModel
FEConnection PROFile     TSPool
```

Figure 39. CEMT transaction: INQUIRE screen

If you type:

P

and press ENTER, you get the following screen:

```
   P
   STATUS: ENTER ONE OF THE FOLLOWING

DEletshipped
DUmp
ENDaffinity
RESet
SEcurity
SHUTdown
SNap
STatistics
```

Figure 40. 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  Gtftrace     TCPIPService
AUXtrace     INttrace     TDqueue
Connection   IRc          TErminal
DB2Conn      Journalname  TRAnsaction
DB2Entry     Line         TRDumpcode
DB2Tran      MODename     TSqueue
DEletshipped MONitor      UOW
DOctemplate  Netname      UOWLink
DSAs         PROCesstype  Vtam
DSName       PROGram      Web
DUmpds       Requestmodel
Enqmodel     STatistics
FEConnection SYDumpcode
FENode       SYStem
FEPool       TAsk
FETarget     TCLass
FIle         TCPIP
                                  SYSID=JOHN APPLID=I

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

Figure 41. 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 118.

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 50).

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 the TLT, the XLT, and the 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 42.

```
?INQ TERM
 STATUS:  COMMAND SYNTAX CHECK
  CEMT  Inquire TErminal()
    < CLass() | ALl >
    < TRansaction() >
    < PRiority() >
    < PAgeable | AUtopageable >
    < Inservice | Outservice >
    < ATi | NOAti >
    < TTi | NOTti >
    < NETname() >
    < ACquired | RELeased >
    < CReate | NOCreate >
    < REMotesystem() >
```

Figure 42. 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 43.

```
  in task
  STATUS:  RESULTS - OVERTYPE TO MODIFY

        *** Sample screen required ***
```

Figure 43. 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 44).

```
  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 44. Terminal settings for a selected terminal

Overtype the displayed settings if you want to make changes. Press ENTER to restore the original panel.

You can view a number of expanded panels in succession by entering '?' or 's' in the space at the beginning of the line for each resource you wish to query. After the first expanded panel, pressing ENTER takes you to the next panel.

Hexadecimal values

Some values such as TSQUEUE and NETUOWID may contain nondisplayable characters, which are shown as periods. You can use PF2 on the expanded panel to see the hexadecimal values of these characters. To switch back to character format, press PF2 again.

Variables

PF5 gives a display of variables, similar to that for the CECI command. Here are some examples of its use.

From a FILE display, you may wish to inquire of a particular DSNAME, which is typically 26 characters and can be up to 44 characters in length. Go to the expanded panel, position the cursor on the DSNAME line, and press PF5. You now have a variable &DSN with the required value. Type I DSN(&DSN) on the command line, and press ENTER.

You may wish to find all the TSQUEUE names beginning with a nondisplayable character—for example, X'FF'. Press PF5 for the variables display. Enter X and 2 to make a variable &X of length 2. Press PF2 to switch to hexadecimal, and overtype '4040' with 'FF5C' ('%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* manual. [1]

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 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* manual.

---

[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 OS/390 Release 3.

CEMT DISCARD commands

The DISCARD command allows you to remove an installed resource definition and its corresponding catalog entry from an active CICS system.

DISCARD does not affect the CICS system definition (CSD) file, and you can therefore reinstate the resource using CEDA INSTALL(resourcename). DISCARD commands require ALTER access to the resource to provide similar security to that for SET commands (which use UPDATE access). You cannot discard resources that are currently in use or are CICS owned resources (beginning with "DFH" or "C").

You can discard the following resource types:

- Autinstmodel
- Connection
- DB2Conn
- DB2Entry
- DB2Tran
| - Doctemplate
| - Enqmodel
- FEPI node
- FEPI pool
- FEPI property set
- FEPI target
- File
- JModel
- Journalname
- Partner
| - Processtype
- Profile
- Program
| - Requestmodel
- TClass
| - TCPIPservice
- TDqueue
- Terminal
- Transaction.
| - TSmodel

(Programming information about FEPI commands is in the *CICS Front End Programming Interface User's Guide*.)

There are two ways to discard a resource:

1. If you know the name of the resource that you want to discard, type:

   ```
   CEMT DISCARD resourcetype(resourcename)
   ```

2. If you do not know the name of the resource, type:

   ```
   CEMT INQUIRE resourcetype
   ```

   and a screen similar to Figure 45 appears. Use the carriage return key to move the cursor to the beginning of the line for the resource or resources you want to discard, and type "D" as shown and then press ENTER.

```
  IN PROG
  STATUS:  RESULTS - OVERTYPE TO MODIFY
 D Prog(ABCDTASK) Len(0002936) Ass Pro Ena Pri    Ced
     Res(000) Use(000000)
   Prog(DFHACP ) Len(0008232) Ass Pro Ena Pri
     Res(001) Use(000002) Any Cex Ful
   Prog(DFHAKP ) Len(0005672) Ass Pro Ena Pri
     Res(000) Use(000001) Bel Cex Ful
   Prog(DFHAMP ) Len(0108432) Ass Pro Ena Pri
     Res(000) Use(000009) Any Cex Ful
   Prog(DFHAPATT) Len(0000632) Ass Pro Ena Sha
     Res(005) Use(000018) Any Cex Ful
   Prog(DFHBRCP ) Len(0000000) Ass Pro Ena Pri
     Res(000) Use(000000) Bel Cex Ful
   Prog(DFHCCNV ) Len(0000000) Ass Pro Ena Pri
     Res(000) Use(000000) Any Cex Ful
 D Prog(ERRTASK ) Len(0000448) Ass Pro Ena Pri
     Res(000) Use(000000)
```

Figure 45. 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

```
►►──CEMT Discard─┬─────────────────────────┬──►◄
                 ├─Autinstmodel(value)──────┤
                 ├─Connection(value)────────┤
                 ├─DB2Conn──────────────────┤
                 ├─DB2Entry(value)──────────┤
                 ├─DB2TRAN(value)───────────┤
                 ├─Doctemplate(value)───────┤
                 ├─Enqmodel(value)──────────┤
                 ├─FENode(value)────────────┤
                 ├─FEPool(value)────────────┤
                 ├─FEPropset(value)─────────┤
                 ├─FETarget(value)──────────┤
                 ├─FIle(value)──────────────┤
                 ├─JModel(value)────────────┤
                 ├─JOurnalname(value)───────┤
                 ├─PArtner(value)───────────┤
                 ├─PROCesstype(value)───────┤
                 ├─PROFile(value)───────────┤
                 ├─PROGram(value)───────────┤
                 ├─Requestmodel(value)──────┤
                 ├─TClass(value)────────────┤
                 ├─TCPIPservice(value)──────┤
                 ├─TDqueue(value)───────────┤
                 ├─TErminal(value)──────────┤
                 ├─TRansaction(value)───────┤
                 └─TSmodel(name)────────────┘
```

Options

AUTInstmodel**(**value**)**

specifies the name of the autoinstall model that is to be removed. The name can be up to 8 characters long. The DISCARD AUTINSTMODEL command removes the installed name of an autoinstall model from the autoinstall terminal model (AITM) table and the CICS catalog.

Connection**(**value**)**

specifies the 4-character identifier of the CONNECTION definition to be discarded. When a connection is removed, all its associated sessions are also removed.

For deletion to be successful:

- A non-remote connection must be out of service.

- A remote connection must have no active sessions.

- For MRO connections, the interregion communications facility (IRC) must be closed. (You can use CEMT SET IRC CLOSED to close it.)

- If the local system is a member of a VTAM generic resource group, APPC synclevel 2 connections must not have any outstanding recovery information.

  Other types of connection *can* be discarded, even if there is recovery work outstanding for them. However, it is recommended that you do not discard them if there is. You can use the INQUIRE

CONNECTION RECOVSTATUS command to check.

In some unusual circumstances, it is possible for deletion of an LU61 connection to fail, even though the connection is out of service, because its sessions are still in-service. If this happens, set the connection INSERVICE, then OUTSERVICE, then reissue the DISCARD command.

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.

DOctemplate**(**value**)**

Specifies the name of the document template to be removed.

Enqmodel**(**value**)**

specifies the 8-character identifier of the resource to be discarded.

FENode

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 DFHJ*nn*, where *nn* is the journal number in the range 1–99.

 Note: Specifying DFHJ01 on this command refers to a user journal, *DFHJ01*.

 Any future reference to the journal name will cause it to be reinstalled using the journal model resource in effect at that time.

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.

| **PROC**esstype**(***value***)**
| Specifies the 8–character name of the Processtype to be
| removed. Only disabled processtypes can be discarded.
| If you are using BTS in a sysplex, it is strongly
| recommended that you use CPSM to remove them.

PROFile**(***value***)**
 specifies the name of the profile that is to be removed. The name can be up to 8 characters long. The DISCARD PROFILE command removes the installed name of a profile from the CICS profile table (PFT) and the CICS catalog.

PROGram**(***value***)**
 specifies the name of the program, map set, or partition set that is to be removed. The name can be up to 8 characters long. The DISCARD PROGRAM command removes the installed name of a program, map set, or partition set from the CICS processing program table (PPT) and the CICS catalog.

| **R**equestmodel**(***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.

| **TCP**ipservice**(***value***)**
| Specifies the name of the TCPIP service to be removed.

TDqueue**(***value***)**
 specifies the name of the transient data queue that is to be removed from a running CICS system. The name can be up to 4 characters long.

 Note: When you discard a transient data queue, an implicit DELETEQ command is issued to free the control intervals used by the queue if the queue is an intrapartition queue. If the queue is a logically recoverable intrapartition queue, a syncpoint is taken after the DELETEQ request, but before the queue definition is discarded. This ensures that the associated control intervals have been freed before the definition is discarded.

TErminal**(***value***)**
 specifies the name of the terminal to be removed. The name can be up to 4 characters long. The DISCARD TERMINAL command removes the installed name of a terminal from the terminal control table (TCT) and the CICS catalog.

 Only terminals that use VTAM as an access method or consoles are eligible for deletion. Terminals that represent sessions cannot be deleted with DISCARD TERMINAL; use the DISCARD CONNECTION command instead.

TRansaction**(***value***)**
 specifies the name of the transaction that is to be removed. The name can be up to 8 characters long. The DISCARD TRANSACTION command removes the installed name of a transaction from the CICS program control table (PCT) and the CICS catalog.

| **TS**model**(***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 begining with DFH.

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 157.

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 163.

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 FENode      PROGram      TSQueue      ◄─┐ Resources
 AUTOinstall  FEPOol      REquestmodel UOW
 AUXtrace     FEPRopset   RRms         UOWDsnfail
 CFdtpool     FETarget    STAtistics   UOWLink
 COnnection   FIle        STReamname   Vtam
 DB2Conn      Gtftrace    SYDumpcode   Web
 DB2Entry     INttrace    SYStem
 DB2Tran      IRc         TAsk
 DEletshipped JModel      TCLass
 DOctemplate  JOurnalname TCPIP
 DSAs         Line        TCPIPService
 DSName       MODename    TDqueue
 DUmpds       MONitor     TErminal
 ENQ          Netname     TRAnsaction
 ENQModel     PArtner     TRDumpcode
 EXci         PROCesstype  TSModel
 FEConnection PROFile     TSPool

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

Figure 46. Sample of the screen following either the INQUIRE or the SET command

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

`INQUIRE PROGRAM`

gives you the status of all programs, and for each program gives its attributes. Full details are given in "CEMT INQUIRE PROGRAM" on page 107.

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(dfhlu0e2)`), the names of all the autoinstall models will be displayed.

Sample screen

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

Figure 47. 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 AUTInstmodel──┬────┬──►◄
                                └─All─┘
```

Displayed fields

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

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, and MVS consoles.

For background information about autoinstall, see the *CICS
Resource Definition Guide*.

Input: Press the Clear key and type `CEMT INQUIRE
AUTOINSTALL` (the minimum abbreviation is `CEMT I AUTO`).

You will see a display screen, a sample of which is shown
below. If you want to change any of the values shown, you
can tab to the highlighted fields and overtype them as
required.

Sample screen

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

Figure 48. CEMT INQUIRE AUTOINSTALL screen

If you place the cursor against the result and press ENTER,
CICS displays an expanded format as shown in Figure 49.

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

Figure 49. The expanded display of an individual entry

```
►►──CEMT Inquire AUTOinstall──►◄
```

Displayed fields

AUTOinstall
    identifies this panel as relating to terminal autoinstall.

COnsoles**(***value***)**
    displays the status of autoinstall for MVS consoles in the
    CICS region. The values are:

    Fullauto
        Autoinstall for consoles is active without any
        support from the autoinstall control program. CICS
        generates the required terminal ID automatically
        without calling the control program, and sets the
        default delete-delay period (60 minutes) for the
        console entry.

    Progauto
        Autoinstall for consoles is active, with the support
        of the autoinstall control program (specified on the
        AIEXIT system initialization parameter). The
        autoinstall control program is required to provide
        the terminal ID and, optionally, a delete-delay
        period.

    Noauto
        Autoinstall for consoles is not active.

    Note: You can change this attribute by overtyping it
        with a different value.

Enablesatus**(***value***)**
    displays the overall status of the terminal autoinstall
    facility, indicating whether CICS is enabled to install
    VTAM terminals and MVS consoles. The values
    displayed are:

    Enabled
        CICS can currently install consoles, or terminals,
        or both. To determine which of these the facility is
        enabled for, check the values returned on the
        MAXREQS and CONSOLES options. Enabled is
        returned for the following conditions:

        Terminals
            MAXREQS not equal 0 and autoinstall
            control program is enabled.
        Consoles
            1. CONSOLES displays FULLAUTO.
            2. CONSOLES displays PROGAUTO and
               the autoinstall control program is
               enabled.

    Disabled
        Neither consoles nor terminals can be
        autoinstalled in CICS. DISABLED is returned for
        the following conditions:

        Terminals
            MAXREQS equal 0, or autoinstall control
            program is disabled.
        Consoles
            1. CONSOLES displays NOAUTO.
            2. CONSOLES displays PROGAUTO but
               the autoinstall control program is
               disabled.

Curreqs**(***value***)**
    displays the number of autoinstall requests that are
    currently being processed. (This is not the number of
    terminals currently installed using autoinstall.)

Maximum**(***value***)**
    displays the largest number of autoinstall requests that
    are allowed to queue at any one time. (This is not the
    largest number of terminals that can be installed using
    autoinstall.)

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

Program**(***value***)**

displays the name of the program that controls the autoinstall process. This is either the CICS-supplied default autoinstall program, DFHZATDX, or a user-written program.

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

CEMT INQUIRE AUXTRACE

Function: Retrieve information about the status of auxiliary trace.

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

INQUIRE AUXTRACE tells you:

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

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

Instead of using the SET command, you can overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see "Overtyping a display" on page 49).

For information on using auxiliary tracing, see the *CICS Problem Determination Guide*. See also the description of the CETR transaction, Chapter 16, "CETR—trace control" on page 225.

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 50. 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).

CEMT INQUIRE AUXTRACE

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 51 on page 58.

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

Figure 51. The expanded display of an individual entry

```
►►──CEMT Inquire AUXtrace──►◄
```

Displayed fields

Auxiliary
identifies this panel as relating to auxiliary trace.

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

STatus**(**value**)**
displays the status of auxiliary tracing in your CICS system. The values are:

Start Auxiliary tracing is in progress (the auxiliary trace data set is open).

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

Stop Auxiliary tracing has stopped, and the trace data set is closed.

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

SWITCHAction**(**value) (input only field**)**
specifies that the trace data set is to be switched. The value is:

SWITCH
Makes an immediate switch from the current auxiliary trace data set to the alternate trace data set.

SWITCHStatus**(**value**)**
displays whether automatic data set switching is to occur when the current auxiliary trace data set becomes full. The values are:

Noswitch
Switching does not occur without operator intervention.

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

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

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

CEMT INQUIRE 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




                                        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 52. CEMT INQUIRE CFDTPOOL screen

```
►►──CEMT Inquire CFdtpool──┬──All───┬──►◄
                           └─(value)─┘
```
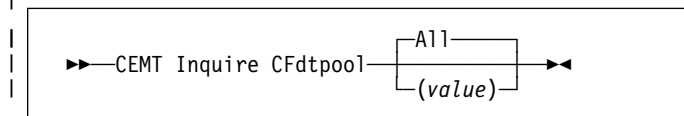
Displayed fields

ALL
     This is the default. CICS displays the names of all
     installed CFDT pools.

CFdtpool(value**)**
     displays information about a named CFDT pool.

     CICS creates a resource entry for a CFDT pool when a
     file resource definition that refers to a coupling facility
     data table is first installed. The resource entry is
     cataloged, and reinstalled from the catalog on a warm or
     emergency restart.

     Note: You cannot discard CFDT pool entries other than
               by a cold or initial start of the CICS region.

Connstatus
     displays the connection status for the named CFDT pool.
     The values are:

     Connected
               The server for the coupling facility data table pool
               is available in this MVS image, and CICS is
               currently connected to it.

     Unconnected
               The server for the coupling facility data table pool
               is available in this MVS image, but CICS is not
               currently connected to it.

     Unavailable
               The server for the coupling facility data table pool
               is currently unavailable in the MVS image.

CEMT INQUIRE 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.
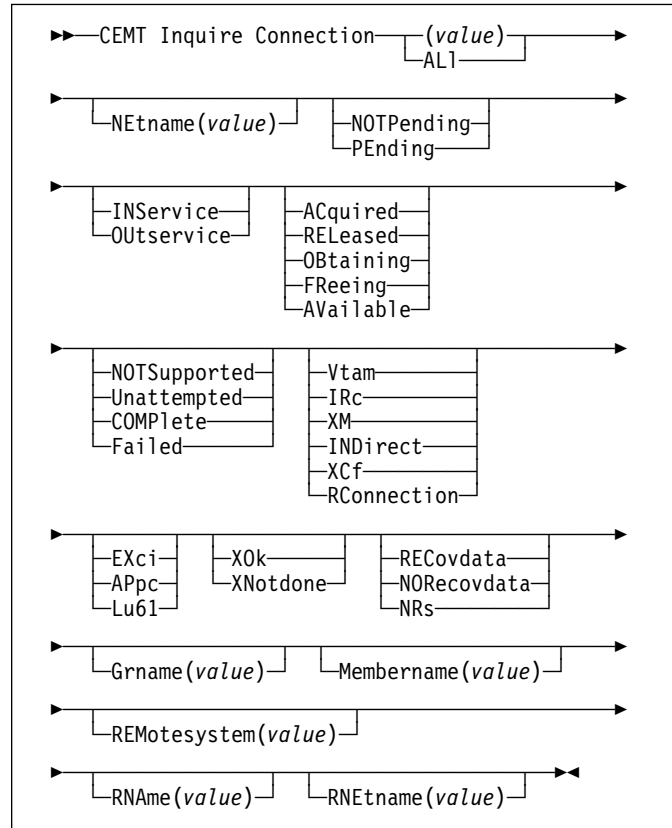
(value**)**
specifies one or more names (1–4 characters) defined for an interregion communication (IRC) or an intersystem communication (ISC) connection.

ALl
is the default.

Sample screen

```
  IN CONN
  STATUS:  RESULTS - OVERTYPE TO MODIFY
    Con(G1GA)                  Ins     Irc Exci
    Con(ISC1) Net(AUHZCIC1)    Ins Acq Vta Appc    Xok Unk
    Con(MRO2) Net(CICSHAA1)    Ins Acq Irc             Unk
    Con(MRO4) Net(CICSHAA2)    Ins Acq Irc             Unk
    Con(MRO5) Net(CICSHAA3)    Ins Acq Irc             Unk
    Con(MRO6) Net(CICSHAB1)    Ins Acq Xcf
```

Figure 53. 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 54 on page 61.

```
  IN CONN
  RESULT - OVERTYPE TO MODIFY
    Connection(MRO5)
    Netname(CICSHAA3)
    Pendstatus( Notpending )
    Servstatus( Inservice )
    Connstatus( Acquired )
    Accessmethod(Irc)
    Purgetype(         )
    Protocol()
    Xlnstatus()
    Recovstatus( Norecovdata )
    Uowaction(          )
    Cqp( Notsupported )
    Grname()
    Membername()
    Affinity(           )
    Remotesystem()
    Rname()
    Rnetname()
```

Figure 54. 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 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.

| **C**qp**(***value***)**
| displays the status of the connection quiesce protocol for the connection. The values are:

| **NOTS**upported
The connection quiesce protocol is not supported for this connection.

| **U**nattempted
CICS has not attempted to use the connection quiesce protocol (because the connection has not been released).

| **COMP**lete
The connection quiesce protocol completed succesfully when the connection was release. The status reverts to UNATTEMPTED when the connection is re-acquired.

| **FA**iled
The connection quiesce protocol failed (for instance, becasue a session failed while the protocol was executing). The status reverts to UNATTEMPTED when the connection is re-acquired.

Grname**(***value) (appc only***)**
displays (for an APPC connection to a generic resource when this system is also a generic resource) the 8-character generic resource name of the connected system. Otherwise this field is blank. CICS assumes that the partner is a generic resource if the two NETNAMEs sent with a BIND are different. Thus, information may be displayed for a partner that is *not* a generic resource but which uses XRF.

Membername**(***value***)**
displays (for an APPC connection to a generic resource when this system is also a generic resource) the 8-character member name (applid) of the connected LU. Otherwise this field is blank. CICS assumes that the partner is a generic resource if the two NETNAMEs sent with a BIND are different. Thus, information may be displayed for a partner that is *not* a generic resource but which uses XRF.

NEtname**(***value***)**
displays the 8-character name by which the remote system is known to the network.

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 OS/390 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 is a pre-CICS Transaction Server for OS/390 system, its performing a cold start has the same effect.) The units of work associated with the connection may need to be individually investigated and forced to commit or backout. Alternatively, the units of work can be unilaterally committed or backed out, according to their associated transaction definitions, by using the SET CONNECTION NOTPENDING command.

If this is an APPC connection, no new syncpoint work (that is, work involving synclevel 2 protocols) can be transmitted across it until a SET CONNECTION NOTPENDING command has been issued. This restriction does not apply to MRO connections.

If you are not concerned by the loss of synchronization caused by the initial (or cold) start of the partner, you can cause the SET CONNECTION NOTPENDING command to be issued automatically by specifying XLNACTION(FORCE) on the CONNECTION definition.

Pendstatus can be reset with Notpending.

For further information about pending units of work, see the *CICS Intercommunication Guide*.

Protocol
displays the protocol in use if this is a VTAM or external CICS interface connection. The values are:

APpc
The connection uses the VTAM LUTYPE6.2 protocol for intersystem communication, or is remote.

EXci The connection uses the external CICS interface for communication between CICS and a non-CICS client program.

Lu61
> The connection uses the VTAM LUTYPE6.1 protocol for CICS-to-CICS or CICS-to-IMS intersystem communication.

Purgetype (*input only field*)
> specifies how associated transactions are to be purged. The values are:

PUrge (*vtam only and irc only*)
> Transactions running on the connected system are abnormally terminated. Transactions are terminated only if system and data integrity can be maintained. A transaction is not purged if its definition specifies SPURGE=NO.

FORCEPurge (*vtam only*)
> All transactions running on sessions on the connected system are immediately terminated abnormally. This can lead to unpredictable results and should be used only in exceptional circumstances.
>
> In some extreme cases (for example, if an error occurs during backout processing), CICS might terminate abnormally.

CAncel
> Automatic initiate descriptors (AIDs) queuing for the specified connection are to be canceled.
>
> AIDs representing scheduled and allocated requests waiting in the local CICS system for the specified connection are canceled. However, TD AIDs with an associated triggered task already started will not be canceled. In addition, the following CICS system AIDs are not purged unless FCANCEL is specified.

Table 1. 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.

FCancel
> All AIDs, including system AIDs, queuing for the specified connection are to be canceled. See Table 1 for a list of those system AIDS that require FCANCEL to remove them. This can lead to unpredictable results and should be used only in exceptional circumstances.
>
> **Note:** FCANCEL does not remove transient data AIDs with an associated triggered task. These aids may be removed by purging the associated task.
>
> An 'AIDS CANCELED' message appears on the CEMT panel whenever AIDs are deleted using the FCANCEL option of the CEMT SET CONNECTION command.

Recovstatus (*appc and mro only*)
> displays whether there is resynchronization work outstanding for this connection. The connection may never have been connected, or may have been quiesced and all resynchronization work completed, or disrupted without quiesce—in which case resynchronization may be necessary. The values are:

NORecovdata
> Neither side has recovery information outstanding.

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

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

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

> Recovstatus can be reset with Norecovdata.

REMotesystem(*value*)
> displays the 4-character name of a connection, if the subject of the inquiry is a remote or an indirect connection. In either case, the named connection can be either a connection entry that links towards the TOR, or an indirect connection that provides the netname of the TOR.
>
> If the subject of the inquiry is an indirect connection, the value of REMOTESYSTEM is taken from the INDSYS option of the CONNECTION definition.
>
> Otherwise this field is blank.

RNAme**(**value**)**

displays the 4-character name by which this connection is known in a remote system, if the subject of the inquiry is a remote connection.

RNEtname**(**value**)**

displays the 8-character netname of the owning TOR, if the subject of this inquiry is a remote connection. If it is blank, but the connection is remote, the system named in the REMOTESYSTEM field has not been installed, and no value was specified for the REMOTESYSNET option when the connection was defined.

Servstatus

displays whether the system can receive and send data. The values are:

INService

The system is in service; that is, it is available for use.

OUtservice

The system is out of service; that is, it is not available for use.

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

Uowaction **(**input-only field**)**

specifies that the normal resynchronization process is to be partially overridden. The value specifies the type of decision to be taken for any units of work that become in-doubt due to a failure of the connection; the decisions are recorded and any data inconsistencies are reported when the connection is next acquired. The values are:

Backout

All UOWs shunted due to the failure of this connection are to be backed out.

COMMit

All UOWs shunted due to the failure of this connection are to be committed.

FORCEUow

All UOWs shunted due to the failure of this connection are to be forced to BACKOUT or COMMIT, as specified on the INDOUBT option of the TRANSACTION definition.

Resync

Any UOWs shunted due to the failure of this connection are to be retried (that is, exchange lognames resynchronization for this connection is to be attempted). This process should normally be started automatically when a connection is acquired or when a UOW is unshunted.

Xlnstatus

displays the status of the exchange lognames (XLN) process. The values are:

XOk **(**appc only**)**

The exchange log names (XLN) process for the APPC connection has completed successfully.

XNotdone **(APPC only)**

The exchange log names (XLN) flow for the APPC connection has not completed successfully. The CSMT log can contain information relating to this state. Synchronization level 2 conversations are not allowed on the connection, but synchronization levels 0 and 1 are still allowed.

No status is displayed if exchange log names is not applicable. This can be because the link:

- Is released
- Is MRO, LU6.1, or single-session APPC
- Does not support synchronization level 2 conversations.

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

CEMT INQUIRE 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 overtype them with the required values.

Sample screen

```
inquire db2conn
STATUS:  RESULTS - OVERTYPE TO MODIFY
  Accountrec( None )           Priority( High )
  Authid(        )             Purgecyclem( 00 )
  Authtype( Userid )           Purgecycles( 30 )
  Comauthid(        )          Signid( CICSHTC1 )
  Comauthtype( Cuserid )       Security(             )
  Comthreadlim( 0001 )         Standbymode( Reconnect )
  Comthreads(0000)             Statsqueue( CDB2 )
  Connecterror( Sqlcode )      Tcblimit( 0012 )
  Connectst( Notconnected )    Tcbs(0000)
  Db2id( DB2A )                Threaderror(N906d)
  Db2release()                 Threadlimit( 0003 )
  Drollback(Rollback)          Threads(0000)
  Msgqueue1( CDB2 )            Threadwait( Twait )
  Msgqueue2(        )
  Msgqueue3(        )
  Nontermrel( Release )
  Plan(        )
  Planexitname( DSNCUEXT )
```

Figure 55. CEMT INQUIRE DB2CONN screen

```
►►──CEMT Inquire DB2Conn──►◄
```

Displayed fields

Accountrec
    returns the minimum amount of DB2 accounting required for transactions using pool threads. The specified minimum 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® connected group name	If the RACF list of group options is not active, DB2 uses the connected group name supplied by the CICS attachment facility as the secondary DB2 authorization ID. If the RACF list of group options is active, DB2 ignores the connected group name supplied by the CICS attachment facility, but the value appears in the DB2 list of secondary DB2 authorization IDs.

        If no RACF group ID is available for this USERID, an 8-character field of blanks is passed to DB2 as the group ID.

Sign The SIGNID parameter of the DB2CONN is used as the resource authorization ID.

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

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

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

Opid
The user operator identification associated with the userid, associated with the CICS transaction, is used as the authorization ID (three characters padded to eight).

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

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

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

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

If no RACF group ID is available for this USERID, an 8-character field of blanks is passed to DB2 as the group ID.

Csign
The SIGNID parameter of the DB2CONN is used as the resource authorization ID.

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

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

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

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

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

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

Comthreads
returns the current number of active command threads.

Comthreadlim
returns the current maximum number of command threads the CICS DB2 attachment allows active before requests overflow to the pool.

Connecterror
returns 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.

Connecting
    CICS is currently attempting to connect to DB2.

Disconning
    CICS is currently disconnecting from DB2.

DB2id
    returns the name of the DB2 subsystem that the CICS
    DB2 attachment is connected to or is to connect to.

DB2release
    returns a four-character value indicating the version and
    release level of the DB2 susbsytem CICS is connected
    to. When CICS is not connected to DB2, blanks are
    returned.

Drollback
    returns a value showing whether the CICS DB2
    attachment is to initiate a SYNCPOINT ROLLBACK if a
    transaction is selected as victim of a deadlock resolution.
    The values are:

    Rollback
        the attachment facility issues a sync point rollback
        before returning control to the application. An SQL
        return code of -911 is returned to the program.

    Norollback
        the attachment facility does not initiate a rollback
        for a transaction. An SQL return code of -913 is
        returned to the application.

Msgqueue1
    returns the name of the first transient data destination to
    which unsolicited messages from the CICS DB2
    attachment are sent.

Msgqueue2
    returns the name of the second transient data
    destination to which unsolicited messages from the CICS
    DB2 attachment are sent.

Msgqueue3
    returns the name of the third transient data destination to
    which unsolicited messages from the CICS DB2
    attachment are sent.

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

    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 subtasks relative to
    the CICS main task (QR TCB). CVDA values are:

    High subtasks attain a higher priority than the CICS
          main task from which the subtask was generated.

    Equal
          subtasks have equal priority with the CICS main
          task.

    Low subtasks have a lower priority than the CICS main
          task.

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

    A protected thread is not terminated immediately when it
    is released. It is terminated only after two completed
    purge cycles, if it has not been reused in the meantime.
    Hence if the purge cycle is set to 30 seconds after it is
    released, a protected thread 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.

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(REBUILD) on
    page173.

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 the current number of subtask TCBs attached to service command, pool, and DB2ENTRY threads.

Tcblimit

returns the current maximum number of subtasks that can be identified to DB2.

Threaderror

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

Abend

| For a second or subsequent SQL error the transaction abends with abend code AD2S, AD2T or AD2U, dependent on the type of error that occurred. The transaction must be terminated and reinitialized before it is allowed to issue another SQL request.

N906D

A transaction dump is taken and the DSNCSQL RMI associated with the transaction is *not* to be disabled. The transaction receives a -906 SQLCODE if another SQL is issued, unless the transaction issues SYNCPOINT ROLLBACK. SYNCPOINT without the ROLLBACK option results in an ASP3 or ASP7 abend. The transaction dump records an abend of AD2S, AD2T or AD2U.

N906

The DSNCSQL RMI associated with the transaction is *not* to be disabled. The transaction receives a -906 SQLCODE if another SQL request is issued, unless the transaction issues a

SYNCPOINT ROLLBACK. SYNCPOINT without the ROLLBACK option results in an ASP3 or ASP7 abend.

Threadlimit

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

Threads

returns the current number of active pool threads.

Threadwait

returns a value showing whether transactions should wait for a pool thread or be abended if the number of active pool threads reaches the threadlimit number. The values are:

Twait

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

Notwait

If all threads are busy, a transaction is terminated with an abend code AD3T.

CEMT INQUIRE DB2ENTRY

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

Figure 56. CEMT INQUIRE DB2ENTRY screen

If you place the cursor against the result and press ENTER, an expanded format is displayed as shown in Figure 57.

```
  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 57. The expanded display of an individual entry

```
►►──CEMT Inquire DB2Entry(data-area)──────────────►

   └─AT(data-area)──┬─START─┬─────────────────────►
                    ├─END───┤
                    └─NEXT──┘

   ┌─ACCOUNTREC(cvda)─┐  ┌─AUTHID(data-area)─┐
   ├─UOW──────────────┤  └───────────────────┘
   ├─TASK─────────────┤
   ├─TXID─────────────┤
   └─NONE─────────────┘

   ┌─AUTHTYPE(cvda)─┐  ┌─DISABLEDACT(cvda)─┐
   ├─GROUP──────────┤  ├─ABEND─────────────┤
   ├─SIGN───────────┤  ├─SQLCODE───────────┤
   ├─TERM───────────┤  └─POOL──────────────┘
   ├─TX─────────────┤
   ├─OPID───────────┤
   └─USERID─────────┘

   ┌─DROLLBACK(cvda)─┐  ┌─ENABLESTATUS(cvda)─┐
   ├─ROLLBACK────────┤  ├─ENABLED────────────┤
   └─NOROLLBACK──────┘  ├─DISABLED───────────┤
                        └─DISABLING──────────┘

   └─PLAN(data-area)─┘

   └─PLANEXITNAME(data-area)─┘

   ┌─PRIORITY(cvda)─┐  └─PROTECTNUM(data-area)─┘
   ├─HIGH───────────┤
   ├─EQUAL──────────┤
   └─LOW────────────┘

   └─PTHREADS(data-area)─┘

   └─THREADLIMIT(data-area)─┘

   └─THREADS(data-area)─┘  ┌─THREADWAIT(cvda)─┐──►◄
                           ├─TWAIT────────────┤
                           ├─NOTWAIT──────────┤
                           └─TPOOL────────────┘
```

Displayed fields

ACcountrec

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

UOW

The CICS DB2 attachment facility causes an accounting record to be produced for each UOW, assuming that the thread is released at the end of the UOW.

TAsk

The CICS DB2 attachment facility causes a minimum of one accounting record for each CICS task to be produced.

A transaction containing multiple UOWs may use a different thread for each of its UOWs (assuming the thread is released at syncpoint). The result may be an accounting record produced for each UOW.

TXid The CICS DB2 attachment facility causes an accounting record to be produced when the transid using the thread changes.

This option applies to DB2ENTRYs that are used by more than one transaction ID. As threads are typically released at syncpoint, a transaction containing multiple UOWs may use a different thread for each UOW. The result may be that an accounting record is produced for each UOW.

NOne
No accounting records are required for transactions using threads from this DB2ENTRY.

DB2 does, however, produce at least one accounting record for each thread when the thread is terminated. Additionally, authorization changes cause accounting records to be produced.

AUTHId
returns an ID to be used for security checking when using pool threads. If an AUTHId is returned, AUTHType is not applicable.

AUTHType
returns the type of ID to be used for security checking for threads on this DB2ENTRY. If an AUTHType is returned, AUTHid is blank. The values are:

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

IDs passed to DB2	How DB2 interprets values
CICS sign-on user ID (USERID)	Represents the primary DB2 authorization ID.
RACF connected group name	If the RACF list of group options is not active, then DB2 uses the connected group name supplied by the CICS attachment facility as the secondary DB2 authorization ID. If the RACF list of group options is active, DB2 ignores the connected group name supplied by the CICS attachment facility, but the value appears in the DB2 list of secondary DB2 authorization IDs.

If no RACF group ID is available for this USERID, then an 8-character field of blanks is passed to DB2 as the group ID.

Sign The SIGNID parameter of the DB2CONN is used as the resource authorization ID.

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

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

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

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

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

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

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

Pool The CICS DB2 attachment facility routes the request to the pool. Message DFHDB2072 is sent to the transient data destination specified by MSGQUEUEn on the DB2CONN for each transaction routed to the pool.

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

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

DRollback
returns returns a value showing whether the CICS DB2 attachment should initiate a SYNCPOINT rollback if a transaction is selected as victim of a deadlock resolution. The values are:

Rollback
The attachment facility issues a sync point rollback before returning control to the application. An SQL return code of -911 is returned to the program.

NORollback
The attachment facility does not initiate a rollback for this transaction. An SQL return code of -913 is returned to the application.

ENAablestatus
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 thread subtasks for this DB2ENTRY relative to the CICS main task (QR TCB). The values are:

High subtasks attain a higher priority than the CICS main task from which the subtask was generated.

Equal

subtasks have equal priority with the CICS main task.

Low subtasks have a lower priority than the CICS main task.

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 DFHtttt, where tttt is the TRANSID.

Sample screen

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

Figure 58. CEMT INQUIRE DB2TRAN screen

If you place the cursor against the result and press ENTER, an expanded format is displayed as shown in Figure 59.

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

Figure 59. The expanded display of an individual entry

```
►►──CEMT Inquire DB2Tran──────────────────────►
                        └─(data-area)─┘
►─────────────────────────────────────────────►
      └─DB2Entry(data-area)─┘
►──────────────────────────────────►◄
      └─Transid(data-area)─┘
```
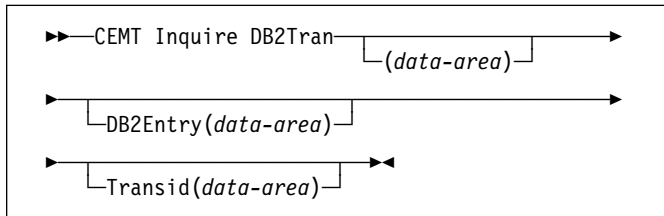
Displayed fields

DB2Entry
> returns the name of the DB2ENTRY to which this DB2TRAN refers. This is the DB2ENTRY with which this additional transaction should be associated.

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 DELETSHIPPED

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

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

To change attributes, you can:

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

Sample screen

```
  IN DELE
  STATUS:  RESULTS - OVERTYPE TO MODIFY
   Del Int( 120000 ) Idl( 020000 )
```

Figure 60. CEMT INQUIRE DELETSHIPPED screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 61.

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

Figure 61. The expanded display

```
►►──CEMT Inquire DEletshipped──────────────────►
►───────────────────────────────────────────►◄
      └─IDle(value)─┘   └─INterval(value)─┘
```
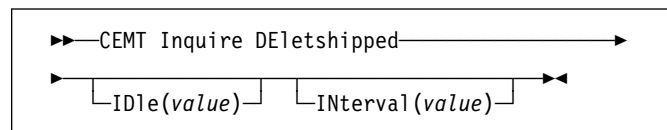
Displayed fields

DEl
> identifies this panel as relating to CICS timeout delete information.

IDle(*value***)**

displays, in the form "**0hhmmss**+", the minimum time that an *inactive* shipped terminal definition must remain installed in this region. When the CICS timeout delete mechanism is invoked, only those shipped definitions that have been inactive for longer than this time are deleted.

The time interval can be in the range 00-99 hours; 00-59 minutes; and 00-59 seconds.

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

INterval(***value***)

displays, in the form "**0hhmmss**+", the interval between invocations of the CICS timeout delete mechanism. The timeout delete mechanism removes any shipped terminal definitions that have not been used for longer than the time displayed by the IDLE option.

The time interval can be in the range 00-99 hours; 00-59 minutes; and 00-59 seconds.

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

| # CEMT INQUIRE 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 DO). You get a display that lists the names and status of
| any document templates. Note that you cannot change any
| of the displayed information.

| **Sample screen**

```
 I DOC
 STATUS: RESULTS - OVERTYPE TO MODIFY
  Doc(test1  ) Tsq Nam(test1        ) App Ebc
     Tem(test1          )

                                 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 62. 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(test1)
     Templatetype(Tsqueue)
     Name(test1)
     Appendcrlf(Append)
     Type(Ebcdic)
     Templatename(test1)
     Ddname()
     Dsname()
```

| *Figure 63. The expanded display of an individual entry*

```
>>──CEMT Inquire DOctemplate────────────────────────────>
                              ├─TSq─────┤
                              ├─TDq─────┤
                              ├─File────┤
                              ├─EXitpgm─┤
                              └─Member──┘

>─────────────────────────────────────────────────────────>
     └─Name(name)─┘  ├─Append───┤  ├─Ebcdic─┤
                     └─Noappend─┘  └─Binary─┘

>────────────────────────────────────────────────────────>
     └─Templatename(name)─┘

>───────────────────────────────────────────────────────><
     └─Templatetype(name)─┘  ├─DDname─┤
                             └─DSname─┘
```
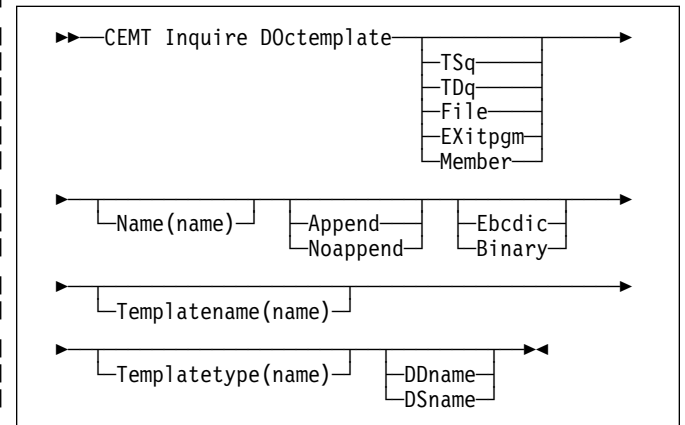
| **Displayed fields**

| **APPENDCRLF**
| returns whether CICS is to delete trailing blanks from
| and append carriage-return line-feed to each logical
| record of the template.

| **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 DFHHTML is used.

| **DOCTEMPLATE**
| returns the name of this DOCTEMPLATE definition.

| **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.

| **MEMBER** a name of the member in the PDS
| described in DDNAME.

| **PROGRAM** a name of a program.

| **TDQUEUE** a name of a TD queue.

| **TSQUEUE** a name of a TS queue.

| **TYPE(BINARY|EBCDIC)**
| 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:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see Figure 64)

- 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 64. CEMT INQUIRE DSAS screen

```
►►──CEMT Inquire DSAs─►◄
```
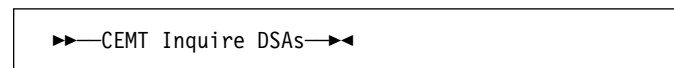
Displayed fields

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.

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.

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.

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.

Udsasize**(***value***)**

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

SOSAbove

displays whether CICS is short on storage in any of the dynamic storage areas above 16MB. The values are:

NOTSOS

CICS is not short on storage in any of the dynamic storage areas above 16MB.

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

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.

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.

CEMT INQUIRE DSNAME

Function: Retrieve information about an external data set.

Description: INQUIRE DSNAME returns information about the status of a data set managed by CICS file control. The data set name is either the name specified on the DSN parameter of the data set's DD statement, or the DSNAME parameter of the data set's file resource definition. Its main purpose is to allow you to monitor failing VSAM data sets. Many of the attributes cannot be determined until a file that references the data set has been opened. If this has not happened, blank values are returned for most attributes.

Note: CEMT has to retrieve QUIESCESTATE information from the ICF catalog. This can slow down processing of this command. QUIESCESTATE is not supplied by default but is retrieved and displayed if requested.

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

- Type `CEMT INQUIRE DSNAME` (the minimum abbreviation is `CEMT I DSN`). You get a display that lists the current status except QUIESCESTATE. To retrieve QUIESCESTATE you can place the cursor at the beginning of the displayed item (in front of the Dsn field) and press the enter key.

- Type `CEMT INQUIRE DSNAME` (`CEMT I DSN`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i dsn vs not`, the resulting display will show you the details of only those external data sets that are VSAM data sets and are not recoverable.

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

```
►►──CEMT Inquire DSName──────────────────────────►

          ┌─ALl─┐
     ├──────────────────────────┤◄
       ─(value)─
          ├─VSam─┤
          └─BDam─┘
       ─FIlecount(value)──
          ├─VALid───┤
          └─INvalid─┘
          ├─Path─┤
          └─BASE─┘
          ├─FWDRECOVAble─┤
          ├─NOTrecovable─┤
          ├─RECOVERAble──┤
          └─UNDetermined─┘
          ├─Static───────┤
          ├─Dynamic──────┤
          └─UNDetermined─┘
       ─FRlog(value)─
          ├─AVailable───┤
          └─UNAvailable─┘
          ├─NOLostlocks──┤
          ├─REMLostlocks─┤
          └─RECOVERLocks─┘
          ├─RETAined───┤
          └─NORetained─┘
          ├─Quiesced───┤
          └─UNQuiesced─┘
       ─BASEDsname(value)──
       └─FWDRECOVLsn(value)─┘
```
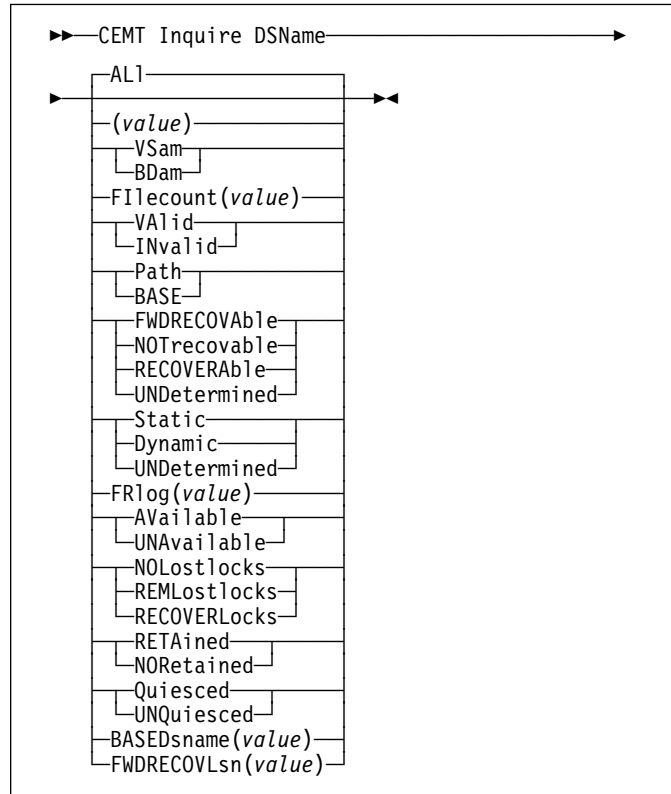
ALl
    is the default.

(value)
    specifies one or more names (1-44 characters) of an external data set.

Sample screen

```
 IN DSN
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Dsn(CICSTS13.CICS.CICSH###.DFHCSD              ) Vsa
     Fil(0001) Val Bas     Sta          Ava
  Dsn(CICSTS13.CICS.CICSHT61.FILEA              ) Vsa
     Fil(0001) Val Bas     Sta          Ava
```

Figure 65. CEMT INQUIRE DSNAME screen

Note: There are blank fields on the screen. With the exception of QUIESCESTATE, these show where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

Because of the large number of keywords supported for this command, only two lines of information for each result are

displayed. The full set of keywords can be displayed by moving the cursor to the beginning of a displayed item (in front of the Dsn field) and pressing the ENTER key. You get a display of all the keyword information for the displayed item, as follows:

```
 IN DSN
 RESULT - OVERTYPE TO MODIFY
   Dsname(CICSTS13.CICS.CICSH###.DFHCSD)
   Accessmethod(Vsam)
   Action(         )
   Filecount(0001)
   Validity(Valid)
   Object(Base)
   Recovstatus(Notrecovable)
   Backuptype(Static)
   Frlog()
   Availability( Available )
   Lostlocks()
   Retlocks(Noretained)
   Uowaction(           )
   Basedsname(CICSTS13.CICS.CICSH###.DFHCSD)
   Quiescestate( Unquiesced )
   Fwdrecovlsn()
```

Figure 66. The expanded display of an individual entry

Displayed fields

DSName**(***value***)**
indicates that this panel relates to an external data set inquiry and displays the 44-character name of an external data set.

Accessmethod**(***value***)**
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.

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.

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.

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.

Recovstatus

displays the recovery status of the data set. The values are:

FWDRECOVAble

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

NOTrecovable

Updates to the data set are not logged.

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

The condition remains until cleared by a CEMT or EXEC CICS SET DSNAME REMOVE command, or by an initial or cold start of CICS (if the associated data set is not in backout-failed state). While the data set is in this state, backout logging is performed for a particular request based on the specification in the file definition. Therefore backout logging may occur for requests via one file and not via another.

RECOVERAble

All updates to the data set are logged for backout.

UNDetermined

The recovery status of the data set is not known because files accessing the data set have not been opened for output.

Backuptype

displays the backup type of the data set. The values are:

Static

The data set is accessed in non-RLS mode and is not eligible for BWO processing. All the files in this data set must be closed and the data set

serialized by a backup manager such as DFSMS/MVS® before a data set backup copy can be made.

Dynamic

The data set is accessed in RLS mode and is eligible for BWO processing. Data set backup is allowed while it is open for output, using a data set backup manager such as DFSMShsm™ or DFSMSdss™. [2]

Notapplic

The data set has not been opened by the CICS region in which the command is issued, or the data set is BDAM or a VSAM PATH. NOTAPPLIC is also returned if the dataset has been opened in RLS mode. The VSAM catalog should be referred to get the BWO status.

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.

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.

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.

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.

---

[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.

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.

Uowaction **(**_input only field_**)**

specifies the action to be taken for all shunted units of work that reference this data set. The values are:

Backout

Specifies that all shunted in-doubt units of work that hold locks on the data set should be backed out.

Commit

Specifies that all shunted in-doubt units of work that hold locks on the data set should be committed. Specifies that all shunted in-doubt units of work that hold locks on the data set should be forced to BACKOUT or COMMIT, as specified by the ACTION attribute on the transaction resource definition.

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.

FWDRECOVLsn**(**_value_**)**

displays the name of the log stream that is used to log the updates if this data set is defined as forward recoverable. CICS returns nulls if the data set is not forward recoverable.

The log stream name returned is either:

- The log stream name specified directly in the ICF catalog, or

- For a non-RLS access mode data set that does not have forward recovery attributes in the ICF catalog, it is a log stream name identified by CICS through a journal name generated from the FWDRECOVLOG value.

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 overtype your changes on the INQUIRE screen after tabbing to the appropriate field. See "Overtyping a display" on page 49.

For information on using dumps, see the *CICS Problem Determination Guide*.

Input: Press the Clear key and type `CEMT INQUIRE DUMPDS` (the minimum abbreviation is `CEMT I DU`).

You will get a display screen, a sample of which is shown below. If you want to change any of the values shown, you can then tab to the highlighted or blank fields and overtype them as required.

Sample screen

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

Figure 67. CEMT INQUIRE DUMPDS screen

Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 68.

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

Figure 68. The expanded display

```
►►──CEMT Inquire DUmpds──►◄
```

Displayed fields

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

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

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* 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 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.
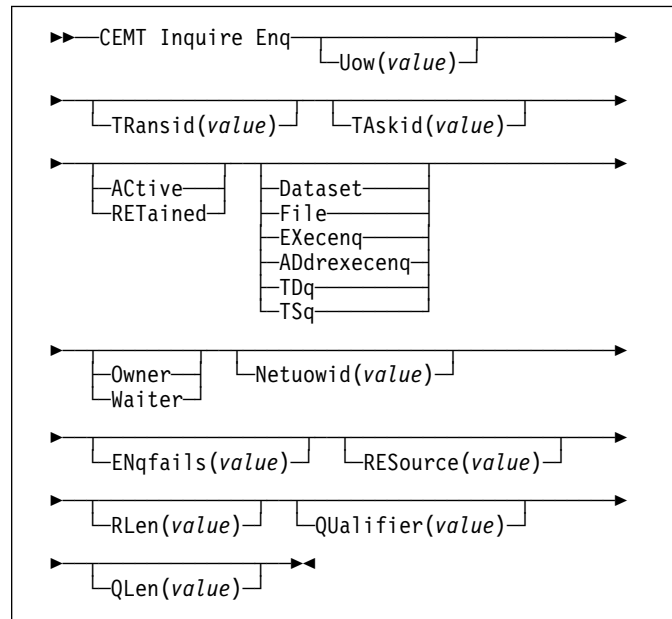
Sample screen

```
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 69. CEMT INQUIRE ENQ screen

Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 70.

```
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 70. The expanded display of an individual entry

| **Displayed fields**

| **D**uration**(***value***)**
| displays the elapsed time (in seconds) since the
| enqueue f entered its current state of OWNER, WAITER
| or RETAINED.

| **ENQF**ails**(***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.

| **ENQS**cope**(***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.

| **N**etuowid**(***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.

| **QU**alifier**(***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 2
| on page 84. If no QUALIFIER data applies (that is for
| EXECENQ, ADDREXECENQ, or TSQ), a value of zero
| is displayed.

| **QL**en**(***value***)**
| displays the length of the data, in the range 0 through
| 255, displayed in the QUALIFIER field.

| **REL**ation
| displays whether the data is associated with the owner
| of the enqueue or with a task waiting for the enqueue.
| The values are:

| **O**wner
| The (*value*), NETUOWID, TASKID, and TRANSID
| are those of the owner of the enqueue.

| **W**aiter
| The (*value*), NETUOWID, TASKID, and TRANSID
| are those of a waiter for the enqueue.

| **RES**ource**(***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 2 on page 84.

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

| **RL**en**(***value***)**
| displays the length of the data, in the range 0 through
| 255, displayed in the RESOURCE field.

| **S**tate
| displays the state that the enqueue is held in. The
| values are:

| **AC**tive
| The enqueue is held in active state.

| **RET**ained
| The enqueue is held in retained state. Its owning
| UOW has been shunted, or is in the process of
| being shunted.

| **TA**skid**(***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.

| **TR**ansid**(***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.

| **TY**pe
| displays the type of resource being enqueued upon. The
| values are:

| **D**ataset
| 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.

| **F**ile 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.

| **EX**ecenq
| The resource is associated with an EXEC CICS
| ENQ request. RESOURCE contains the enqueue
| argument passed on the request.

| **AD**drexecenq
| 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)

| **TD**q 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.

| **TS**q 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 2.

Table 2. 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

| **U**ow**(***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 overtype them as required.

| **Sample screen**

```
  INQ ENQMODEL
  STATUS:  RESULTS - OVERTYPE TO MODIFY
   Enqm(*      )                                NOT FOUND
```
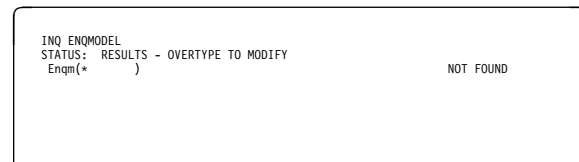
| *Figure 71. CEMT INQUIRE ENQMODEL screen*

| **Note:** There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

```
►►──CEMT Inquire ENQModel────────────────►

►───┬─(value)──────────┬──┬─────────┬──►◄
    ├─ENQSCOPE(value)──┤  ├─DISABLED─┤
    └─ENQNAME(value)───┘  ├─ENABLED──┤
                          └─WAITING──┘
```
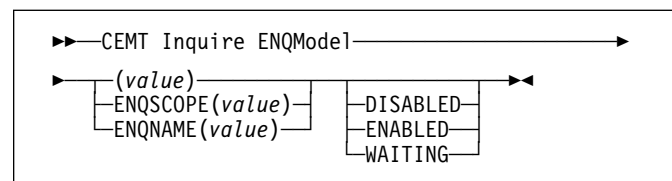
| **Displayed fields**

| **ENQMODEL**
| The 8-character identifier of the resource definition

| **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.

| **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.

| **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.

```
►►──CEMT Inquire Exci─────────────────────►◄
                        ┌─Task(number)─┐
|                       └─Urid(value)──┘
```
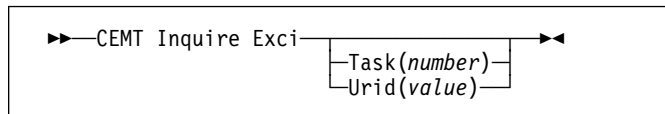
Sample screen

```
 INQ EXCI
 STATUS: RESULTS
  Exc(HADBS1..CICSWK1 - MV29          ) Tas(0000121)
  Exc(AHCLN2..TRNSL 1 - MV29          ) Tas(0000125)
     Uri(B 67B1F77EB7D      21 1 1    )
```

Figure 72. CEMT INQUIRE EXCI screen

Note: There are blank fields on the screen where a value
does not apply or is 'negative'; that is, the value
begins with 'No'. To modify these fields, locate them
by tabbing (they appear in the same sequence as in
the expanded format), and overtype with input valid
for that field. You may find it more convenient to use
the expanded format when setting one of these
values, (negating is easier because you are
overtyping a displayed field with its negative causing
the result to become non-displayed).

If you place the cursor against a specific entry in the list and
press ENTER, CICS displays an expanded format as shown
in Figure 73.

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

Figure 73. 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.

| **U**rid**(***value***)**
| displays, when the job is using RRMS to coordinate
| updates, and when there is an active DPL request for
| the session, a 32-character string containing the
| hexadecimal representation of the RRMS Unit of
| Recovery Identifier.

CEMT INQUIRE FEPI

Comments: For information on CEMT INQUIRE FECONNECTION, CEMT INQUIRE FEPOOL, CEMT INQUIRE FEPROPSET, and CEMT INQUIRE FETARGET, see the *CICS Front End Programming Interface User's Guide*.

CEMT INQUIRE FILE

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 overtype them with the required values.

```
►►──CEMT Inquire FIle──┬──────────AL1──────────┬──►
                       └──(value)──┘

►──┬─Vsam───┬──┬─OPen───┬──┬─ENabled───┬──►
   ├─BDam───┤  └─CLosed──┘  ├─UNenabled─┤
   └─REMote─┘               └─DIsabled──┘

►──┬─REAd───┬──┬─UPdate───┬──┬─ADdable───┬──►
   └─NORead─┘  └─NOUpdate─┘  └─NOAddable─┘

►──┬─BRowse───┬──┬─DElete───┬──►
   └─NOBrowse─┘  └─NODelete─┘

►──┬─EXclusive───┬──┬─OLd───┬──┬─RLs────┬──►
   └─NOEXclusive─┘  └─Share─┘  └─NOTRls─┘

►──┬─EMptyreq───┬──┬─DSname(value)─┬──►
   └─NOEMptyreq─┘

►──┬─NOTTable──┬──┬─LOAd───┬──►
   ├─CIcstable─┤  └─NOLoad─┘
   ├─USertable─┤
   └─CFTable───┘

►──┬─CFDtpool(value)─┬──►

►──┬─Tablename(value)─┬──┬─LOCKING────┬──►
                        └─COntention─┘

►──┬─Maxnumrecs(value)─┬──►

►──┬─Keylength(value)─┬──►

►──┬─RECordsize(value)─┬──►◄
```
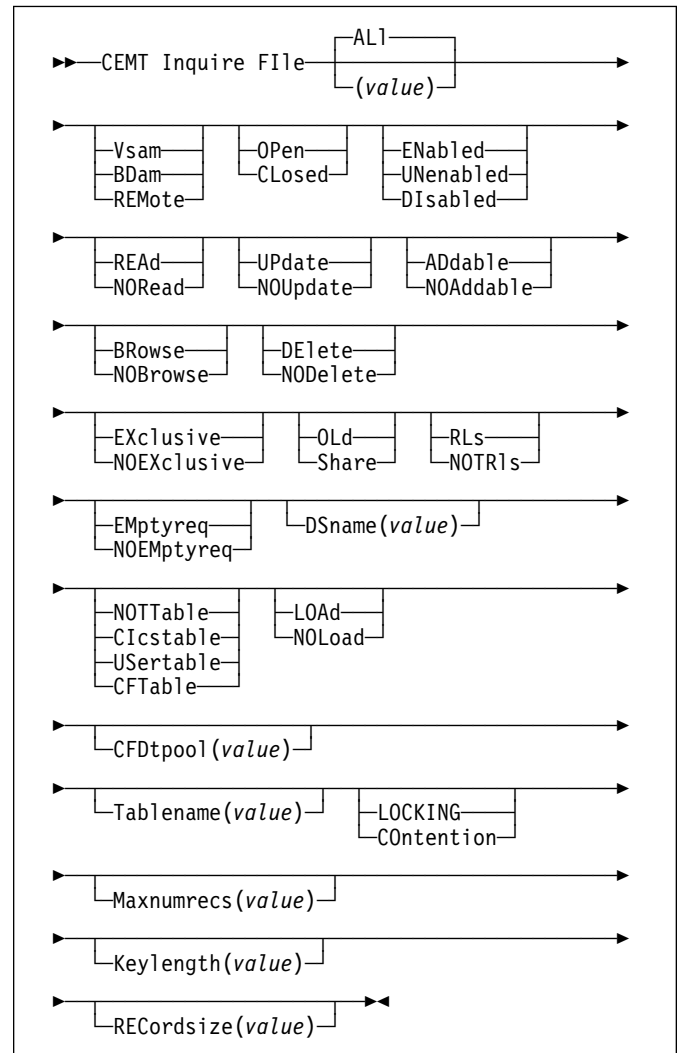
(value**)**

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

ALl

is the default. Information about all files is given, unless you specify a selection of files to be queried.

Sample screen

```
 IN FILE
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Fil(DFHCMACD) Vsa Ope Ena Rea                   Sha
       Dsn( CICSTS13.CICS.DFHCMACD                    )
 Fil(DFHCSD  ) Vsa Clo Une Rea Upd Add Bro Del    Sha
       Dsn( CICSTS13.CICS.CICSH.DFHCSD                )
 Fil(DFHDBFK ) Vsa Clo Ena Rea Upd Add Bro Del    Sha
 Fil(DFHLRQ  ) Vsa Ope Ena Rea Upd Add Bro Del    Sha
       Dsn( CICSTS13.CICS.CICSH.DFHLRQ              )
 Fil(FILEA   ) Vsa Clo Ena Rea Upd Add Bro Del    Sha
       Dsn( CICSTS13.CICS.CICSH.FILEA                ) Cft Loa
```

Figure 74. CEMT INQUIRE FILE screen

Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them

by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 75.

```
  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( CICSTS13.CICS.CICSH.DFHCSD            )
    Table( Nottable )
    Loadtype( Noload )
    Cfdtpool(          )
  + Tablename(          )
    Updatemodel(          )
    Maxnumrecs( 00000000 )
    Keylength( 000 )
    Recordsize( 00000 )
```

Figure 75. 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.

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.

| **K**eylength(*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.

| **L**oadtype
| 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.

| **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:

| **REA**d | You can read records in the file. |
| **NOR**ead | You cannot read records in the file. |

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

RECordsize**(**value**)**
displays the actual size of fixed-length records, or the maximum size of variable-length records.

If the file is not open, CICS displays the value specified in the installed file definition. If the file definition does not specify a record size, CICS displays a zero value.

Generally, if a file is open, CICS obtains the record size from the associated data set, in which case CICS displays the value from the data set. However, files that refer to coupling facility data tables defined with LOAD(NO) must get their record size from the file definition, in which case CICS returns the defined value when the file is open. This value must also match that of the coupling facility data table if it has already been created.

Rlsaccess **(**VSAM only**)**
displays whether the file is defined to be accessed in RLS mode. The values are:

RLs   The file is defined to be accessed in RLS mode.

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

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

Table
displays whether the file name specified on the FILE parameter represents a data table. The values are:

NOTTable
The file name does not represent a data table.

CFTABLE
The file name represents a coupling facility data table.

CICSTABLE
The file name represents a CICS-maintained data table.

USERTABLE
The file name represents a user-maintained data table.

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

Tablename**(**value**)**
displays the table name specified for the coupling facility data table on the file resource definition, if one is specified, or displays the file name if no table name was specified on the file resource definition.

CICS displays blanks if the file does not refer to a coupling facility data table

Updatemodel **(CFDT only)**
displays the update model specified for the coupling facility data table in the installed file definition. The values are:

| CONTENTION
| The coupling facility data table is updated using
| the contention model.
| LOCKING
| The coupling facility data table is updated using
| the locking model.

Updatestatus
    displays whether the file is updatable. The values are:

UPdate
        You can update records in the file.

NOUpdate
        You cannot update records in the file.

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

CEMT INQUIRE GTFTRACE

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 overtype it as required.

Sample screen

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

Figure 76. CEMT INQUIRE GTFTRACE screen

If you place the cursor against the result and press ENTER,
CICS displays an expanded format as shown in Figure 77.

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

Figure 77. The expanded display

```
►►──CEMT 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.

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

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 Clo
```

Figure 78. CEMT INQUIRE IRC screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 79.

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

Figure 79. The expanded display

```
►►──CEMT Inquire IRc──►◄
```

Displayed fields

IRc
indicates that this panel relates to an IRC inquiry.

Openstatus**(**value**)**
displays whether the interregion communication (IRC) facility is open or closed. The values are:

Closed
Interregion communication for this CICS system is terminated after tasks using the session have terminated normally.

Open
Interregion communication for this CICS system is initialized. This allows connections to be established between this system and other systems. You can control the availability of individual connections with the INSERVICE or OUTSERVICE settings on the connection names,

using CEMT INQUIRE CONNECTION or CEMT SET CONNECTION.

The DFHIR3780 message is issued to the console log if the region made an unsuccessful attempt to start IRC.

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.
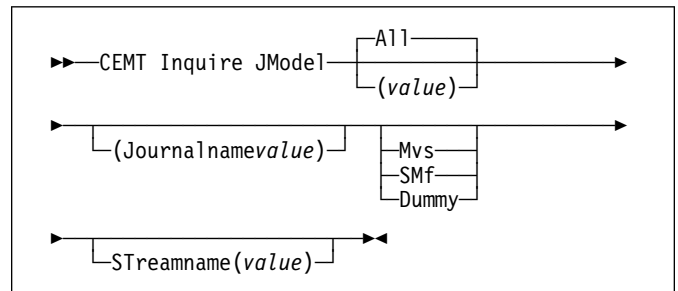
(value**)**
  specifies one or more journal models.

All   is the default.

Sample screen

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

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 80 on page 94.

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

Figure 80. The expanded display of an individual entry

Displayed fields

JModel**(**value**)**

indicates that this panel relates to a journal model
inquiry and displays the 8–character name of an
installed journal model.

Journalname

displays the 8–character journal name, which can be a
specific or a generic name, for which this journal model
can be used.

STreamname

displays the MVS log stream name (LSN) associated
with this journal model. The name can be a specific
| LSN, or a template using any 3 of the 4 symbols
| &USERID, &APPLID, &JNAME and &SYSID.

An LSN or template can be up to 26 characters in
length.

Type

displays the log stream type. The possible types are:

Mvs Records for journals defined with this journal
model are written to an MVS log stream.

SMf Records for journals defined with this journal
model are written to the MVS SMF data set.

Dummy

Records for journals defined with this journal
model are not written to any log stream.

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

CEMT INQUIRE JOURNALNAME

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 JO). You get a display that lists
  the current status.

- Type CEMT INQUIRE JOURNALNAME (CEMT I JO) followed by
  as many of the other attributes as are necessary to limit
  the range of information that you require. So, for
  example, if you enter cemt i jo sm, the resulting display
  will show you the details of only those logs that are of
  SMF log stream type.

```
►►──CEMT Inquire JOurnalname──┬──All──┬──────────────►
                              └─(value)─┘

►──┬────────┬──┬──────────┬──────────────────────────►
   ├──Mvs───┤  ├─Enabled──┤
   ├──SMf───┤  ├─DIsabled─┤
   └──DUmmy─┘  └─FAiled───┘

►──┬──────────────────────┬──►◄
   └─STreamname(value)─────┘
```
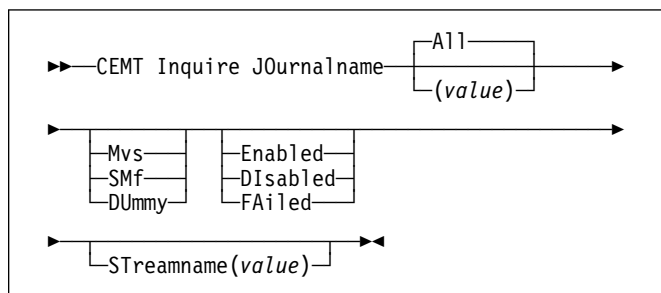
(value**)**

specifies one or more journal names.

To inquire on journals defined with a numeric identifier in
the range 1–99, specify journal name DFHJ*nn*, where *nn*
is the journal number.

To inquire on the system log, specify DFHLOG.

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

All is the default.

Sample screen

```
   IN JOUR
   STATUS: RESULTS – OVERTYPE TO MODIFY
    Journaln(DFHJ02  ) Smf Ena
    Journaln(DFHLGLOG) Mvs Ena
       Str(CICSHT##.CICSHTA1.DFHLGLOG )
    Journaln(DFHLOG  ) Mvs Ena
       Str(CICSHT##.CICSHTA1.DFHLOG   )
    Journaln(DFHSHUNT) Mvs Ena
       Str(CICSHT##.CICSHTA1.DFHSHUNT )
```

Figure 81. 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 82 on page 95.

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

Figure 82. 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 a SET JOURNALNAME command, or until following the next CICS restart. The log stream should be deleted from the MVS system logger inventory before being used again.

Note: You can reset STATUS with ENABLED or DISABLED.

STreamname**(**value**)**

displays the MVS log stream name associated with the journal. CICS displays 8 blanks (X'40') if the journal is defined with a type of DUMMY or SMF.

Type

displays the type of log stream. The possible types are:

Mvs   Records for journals defined with this journal model are written to an MVS log stream.

SMf   Records for journals defined with this journal model are written to the MVS SMF data set.

DUmmy

Records for journals defined with this journal model are not written to any log stream.

CEMT INQUIRE 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 49), but you cannot use the symbols * and + to specify a family of identifiers. You can then tab to the highlighted or blank fields and overtype them with the required values.

```
►►──CEMT Inquire Line──(value)─────────────►

 ►──┬──────────────┬──┬──────────────┬──►◄
    └─NUmber(value)─┘  └─NEgpoll(value)─┘

 ►──┬─Inservice──┬──►◄
    └─Outservice─┘
```
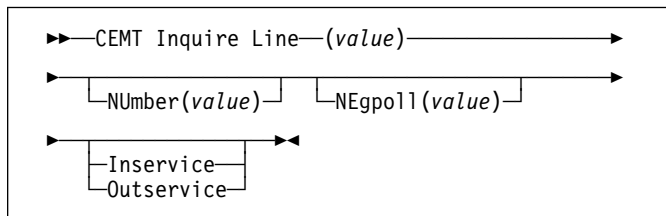
Sample screen

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

Figure 83. CEMT INQUIRE MODENAME screen

Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against a specific entry in the list, CICS displays an expanded format as shown in Figure 84.

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

Figure 84. The expanded display

Displayed fields

Inservice
  The line is available for use. Changing the status of the line does not affect the equivalent status of the terminals on it.

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.

NEgpoll**(***value***)**
  displays the negative poll delay interval (0–20000 milliseconds). For a TCAM line, this specifies the interval before control is passed to the terminal error program (DFHTEP) when a CICS task is not ready to accept a record from an input process queue.

  If the CICS task issues a read before the interval expires, processing continues normally and DFHTEP is not notified. The default value is zero.

NUmber**(***value***)**
  displays the relative line number of a pooled line. 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.

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 overtype them with the required values.
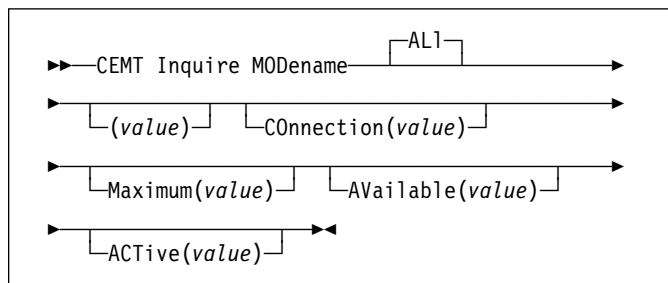
(*value***)**
    specifies one or more names (1-8 characters) of a group of sessions.

ALl
    is the default.

Sample screen

```
  I MODE
  STATUS: RESULTS - OVERTYPE TO MODIFY
   Mod(SNASVCMG) Con(C  ) Max(002) Ava( 000 ) Act(000)
   Mod(MODE1  ) Con(C  ) Max(001) Ava( 000 ) Act(000)
```

Figure 85. CEMT INQUIRE MODENAME screen

Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against a specific entry in the list, CICS displays an expanded format as shown in Figure 86.

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

Figure 86. 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 Off                   Fre( 000000 ) Gmt Sub(IYAH)
```

Figure 87. 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 88.

```
 IN MON
RESULT - OVERTYPE TO MODIFY
  Monitor
  Status( Off )
  Perfclass( Noperf )
  Eventclass( Noevent )
  Exceptclass( Noexcept )
  Conversest( Noconverse )
  Syncpointst( Nosyncpoint )
  Frequency( 000000 )
  Time(Gmt)
  Subsystemid(CICS)
```

Figure 88. The expanded display

```
►►──CEMT Inquire MONitor─────────────────────►
                            ├─Gmt───┤
                            └─Local─┘

►────────────────────────────►◄
          └─SUbsystemid(value)─┘
```
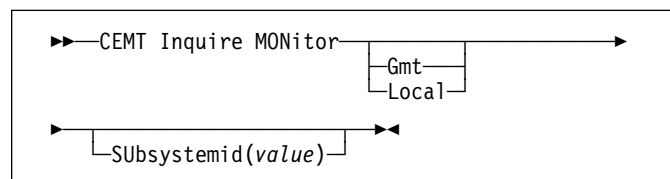
Displayed fields

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 will be produced.

Noconverse
Separate performance class records will not be produced.

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

Eventclass
displays whether the SYSEVENT class of monitoring data is being collected. The values are:

Event
SYSEVENT-class monitoring data is being collected.

Note: There are implications with using the Event class of monitoring and MVS 5.1 WLM. For information, see the *CICS Performance Guide*.

Noevent
SYSEVENT-class monitoring data is not being collected.

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.

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 001500–240000.

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.

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 monitoring SYSEVENT class records. The subsystem identification defaults to the first 4 characters of the VTAM generic APPLID.

For information on the SYSEVENT class of monitoring data and the VTAM generic APPLID, 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).

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* manual.

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.
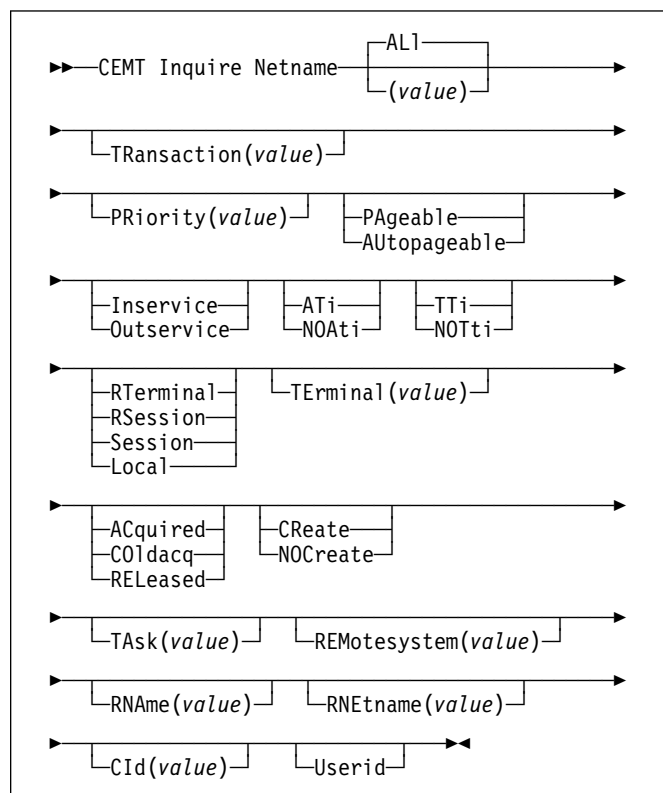
(value**)**
> specifies an 8-character network name.

ALl
> is the default.

Sample screen

```
IN NETNAME
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Net(IGKS205 ) Tra(CEMT) Pri( 000 ) Pag Ins Ati Tti
    Ter(S205) Acq
```

Figure 89. 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 90.

```
IN NET
RESULT - OVERTYPE TO MODIFY
  Netname(IG2S944D)
  Transaction(CEMT)
  Priority( 000 )
  Pagestatus( Pageable )
  Servstatus( Inservice )
  Atistatus( Ati )
  Ttistatus( Tti )
  Nature(Local)
  Purgetype(           )
  Terminal(944D)
  Termstatus( Acquired )
  Createsess( Nocreate )
  Task(00033)
  Remotesystem()
  Rname()
  Rnetname()
  Cid()
```

Figure 90. 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.

- 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 owning region.

Pagestatus

displays whether pages after the first in a series are written to the terminal on request from the operator or automatically. The values are:

PAgeable

Pages are written on request.

AUtopageable

Pages are written automatically.

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

PRiority**(***value***)**

displays a 3-character string indicating the priority of this terminal relative to other terminals. The priority of a task is the sum of the transaction priority, the terminal priority, and the operator priority. Priority has no meaning for terminals that are ISC sessions being used as alternative facilities. The value is in the range 0–255, where 255 is the highest priority.

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

Purgetype **(***input only field***)**

specifies whether the transactions running with the named terminal are to be purged. The values are:

PUrge

Transactions are to be terminated only if system and data integrity can be maintained. A transaction is to be purged if its definition specifies SPURGE=NO.

Forcepurge

Transactions are to be purged immediately. This can lead to unpredictable results and should be used only in exceptional circumstances.

REMotesystem**(***value***)**

displays the 4-character name of a connection, if the subject of the inquiry is a remote terminal. The named connection can be either a connection entry that links towards the terminal-owning region (TOR), or an indirect connection that provides the netname of the TOR.

Otherwise this field is blank.

RNAme**(***value***)**

displays the 4-character name by which this terminal is known in the system in which it is defined (the TOR). RNAME applies only to terminals defined as remote; for others, blanks are displayed.

RNEtname**(***value***)**

displays the 8-character netname of the owning TOR, if the subject of the inquiry is a remote terminal. If this field is blank **and** the terminal is remote, the system named in the REMOTESYSTEM field has not been installed, and no value was specified for the REMOTESYSNET option when the terminal was defined.

Servstatus

displays whether the terminal is available for use. The values are:

Inservice

The terminal is available for use. For VTAM, INSERVICE means that the terminal can be ACQUIRED. For IRC sessions, INSERVICE means that the connection to the MRO partner is INSERVICE.

Outservice

The terminal is not available for use. Setting a terminal OUTSERVICE means that the terminal can no longer be used by transactions. If PURGE or FORCEPURGE is also specified, any transaction using the terminal is terminated abnormally. If PURGE or FORCEPURGE is not specified, the transaction is allowed to terminate normally, but no further transactions are allowed to use the terminal. For VTAM, setting a terminal OUTSERVICE also causes it to be released and the operator to be signed off, either immediately or when the current transaction has terminated.

For IRC sessions, OUTSERVICE means that the connection to the MRO partner is OUTSERVICE.

In an LU6.1 ISC session, the connection is set OUTSERVICE if there are no allocatable sessions left.

Note: For physical terminals and LU6.1 sessions, you can reset this value by overtyping it with a different value.

TAsk**(***value***)**
displays the task number of the transaction that is running on this terminal.

TErminal**(***value***)**
displays a 4-character terminal identifier (1-4 characters) as specified in an installed terminal definition.

Termstatus **(***vtam only***)**
displays whether CICS is in session with the logical unit represented by this terminal. The values are:

ACquired
CICS is in session with the logical unit represented by the terminal.

RELeased
CICS is not in session with the logical unit represented by the terminal. If you set this option to RELEASED, a session is terminated immediately if you also specify the PURGE option, otherwise the session is terminated when the current active transaction finishes.

Note: For physical terminals and LU6.1 sessions, you can reset this value by overtyping it with a different value or with:

COldacq
This is a special form of ACQUIRED, where no resynchronization is required. If the previous session abended, the use of COLDACQ overrides CICS integrity control. This 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.

```
         ┌────All────┐
►►─CEMT Inquire PArtner─┤           ├─────────────►
         └──(value)──┘

►─┬─────────────────┬──┬──────────────────┬──────►
  └─NETWork(value)──┘  └─NETName(value)───┘

►─┬─────────────────┬──┬──────────────────┬──────►
  └─Profile(value)──┘  └─Ltpname(value)───┘

►─┬──────────────────┬──────────────────────────►◄
  └─Tpname(value)────┘
```
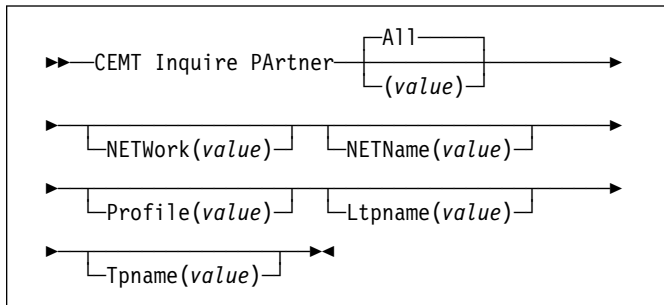
(value**)**
   specifies the 8-character partner name as specified in the installed partner definition.

All   is the default.

Sample screen

```
 I PART
 STATUS:  RESULTS
   Par(IST1ISC2)           Netn(IYMVM2  ) Pro(DFHCICSA)
      Ltp(04) Tpn(IST1              )
   Par(IST2ISC2)           Netn(IYMVM2  ) Pro(DFHCICSA)
      Ltp(04) Tpn(IST2              )
```

Figure 91. 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 92.

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

Figure 92. The expanded display of an individual entry

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 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:

| • Overtype your changes on the INQUIRE screen after
| tabbing to the appropriate field.

| • Use the CEMT SET PROCESSTYPE command.

```
 ►►─CEMT Inquire PROCesstype─┬─AL1──┬─────►
                             └─(value)─┘

 ►─┬────────────┬─┬──────────────┬─►
   └FIle(value)─┘ └AUditlog(value)─┘

 ►─┬─ACtivity─┬─┬─Enabled──┬─►◄
   ├─FUll─────┤ └─Disabled─┘
   ├─Off──────┤
   └─Process──┘
```
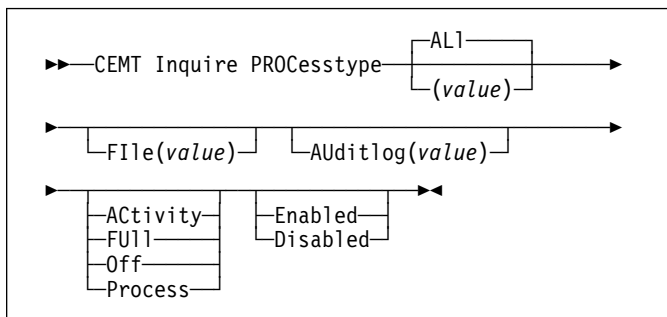
| **AL**l
| is the default. Information about all process-types is
| displayed, unless you specify a selection of
| process-types to be queried.

| *(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 93. 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 94.

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

| *Figure 94. 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:

| **AC**tivity
| Activity-level auditing. Audit records are written
| from:

| 1. The process audit points
| 2. The activity primary audit points.

| **FU**ll   Full auditing. Audit records are written from:

| 1. The process audit points
| 2. The activity primary *and* secondary audit
| points.

| **O**ff   No audit trail records are written.

| **P**rocess
| 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.

| **AU**ditlog*(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:

| **D**isabled
| The installed definition of the process-type is
| disabled. New processes of this type cannot be
| defined.

| **E**nabled
| The installed definition of the process-type is
| enabled. New processes of this type can be
| defined.

| **FI**le*(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.

| **PROC**esstype*(value)*
| indicates that this panel relates to a PROCESSTYPE
| inquiry and displays the 8-character name of a
| process-type.

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.

```
                               ┌─All─┐
►►──CEMT Inquire PROFile───────┼─────┼───►◄
                               └─(value)─┘
```
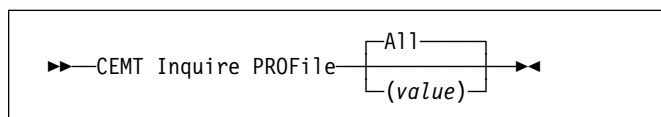
(value**)**
     is the name of a specific profile.

All   is the default.

Sample screen

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

Figure 95. 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 overtype them with the required values.

```
►►──CEMT Inquire PROGram──┬──────ALl──────┬──►
                          ├────(value)────┤
                          └──CLass(value)─┘

►──┬──────────────────┬──┬─PLi───────┬──►
   └──LENgth(value)───┘  ├─CObol─────┤
                         ├─C─────────┤
                         ├─LE370─────┤
                         ├─ASsembler─┤
                         ├─NOTDEfined┤
                         └─JAVA──────┘

►──┬─PROgram──────┬──┬─Enabled──┬──►
   ├─Mapset───────┤  └─DIsabled─┘
   └─PArtitionset─┘

►──┬─PRIvate─┬──────────────────────────────►
   └─Shared──┘  ├─REQuired────┤  ┌─CEDf───┐
               └─NOTRequired─┘  └─NOCedf─┘

►──┬─DYnamic────┬──┬──────────────┬──►
   └─NOTDYnamic─┘  └─REScount(value)─┘

►──┬──────────────────┬──┬─ANy───┬──►
   └─USecount(value)──┘  └─Below─┘

►──┬─CEXeckey─┬──┬─DPlsubset─┬──►
   └─UExeckey─┘  └─Fullapi───┘

►──┬─Quasirent──┬──┬─REMotesystem(value)─┬──►
   └─Threadsafe─┘

►──┬─JVM───────┬──┬─HOtpool────┬──►
   ├─LE370─────┤  └─NOTHotpool─┘
   ├─NATIVE────┤
   └─NOTKNOWN──┘

►──┬─DEBug───┬──┬─JVMClass(value)─┬──►◄
   └─NODebug─┘
```
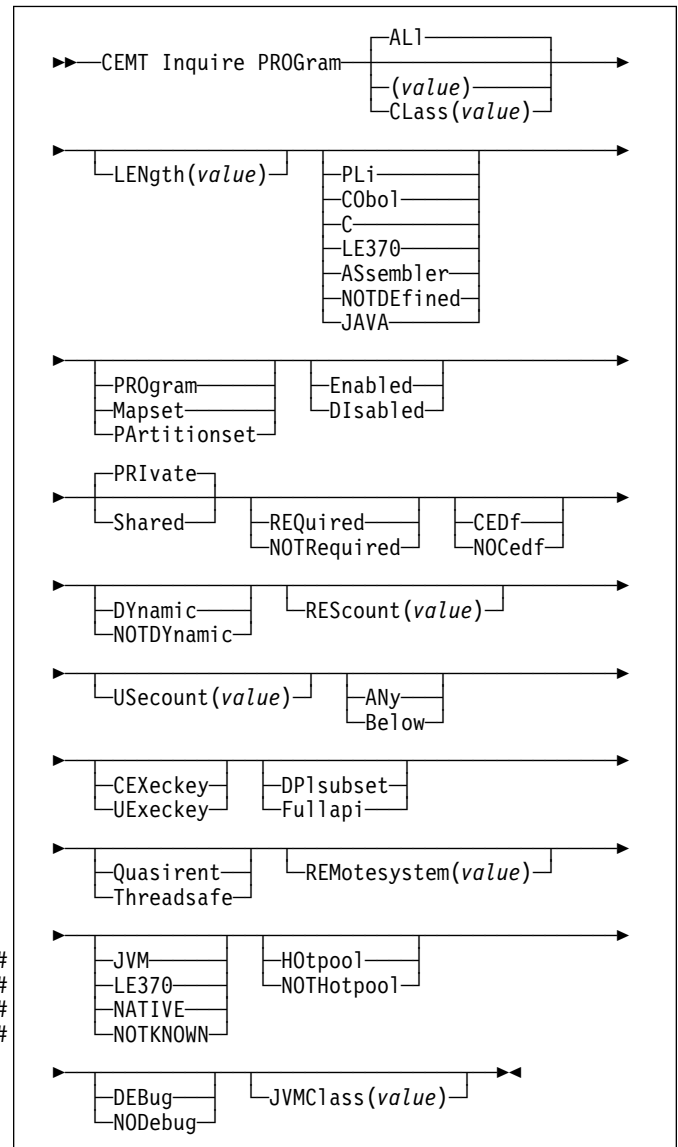
(value**)**
is a program identifier (1–8 characters), which is the name of a specific program entry in the table of installed program definitions. If you omit the program name, the ALL option is assumed by default.

You cannot specify the CLASS option and a program name.

ALl
is the default.

CLass**(**value**)**
is the 2-character suffix of a program list table (PLT). Use this option to inquire on those programs defined in the specified PLT. Note that, unlike a shutdown PLT, it is not mandatory to define a program definition for a startup PLT. However, if you want to use the CLASS option, you must define and install a program resource definition for the PLT specified by "clasid".

If the clasid specified is the suffix of a PLT that does not have a program definition, the INQUIRE PROGRAM CLASS(clasid) command returns a 'CLASS NOT FOUND' message.

Sample screen

```
  IN PROG
  STATUS:  RESULTS - OVERTYPE TO MODIFY
   Prog(CNCLTASK) Len(0000000) Cob Pro Ena Pri    Ced
     Res(000) Use(0000000) Bel Uex Ful
   Prog(DFH$AALL) Len(0002936) Ass Pro Ena Pri    Ced
     Res(000) Use(0000000) Bel Uex Ful
   Prog(DFH$ABRW) Len(0002520) Ass Pro Ena Pri    Ced
     Res(000) Use(0000001) Bel Uex Ful
   Prog(DFH$ACOM) Len(0001128) Ass Pro Ena Pri    Ced
     Res(000) Use(0000000) Bel Uex Ful
   Prog(DFH$AGA ) Len(0000520)     Map Ena Pri
     Res(000) Use(0000002)
   Prog(DFH$AGB ) Len(0000448)     Map Ena Pri
     Res(000) Use(0000000)
   Prog(DFH$AGC ) Len(0000512)     Map Ena Pri
     Res(000) Use(0000000)
   Prog(DFH$AGD ) Len(0000576)     Map Ena Pri
     Res(000) Use(0000000)
 + Prog(DFH$AGK ) Len(0000392)     Map Ena Pri
     Res(000) Use(0000000)
```

Figure 96. 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 97 and Figure 98.

```
  IN PROG
  RESULT - OVERTYPE TO MODIFY
   Program(DFH$AALL)
   Length(0002936)
   Language(Assembler)
   Progtype(Program)
   Status( Enabled )
   Sharestatus( Private )
   Copystatus( Notrequired )
   Cedfstatus( Cedf )
   Dynamstatus(Notdynamic)
   Rescount(000)
   Usecount(0000000)
   Dataloc(Below)
   Execkey(Uexeckey)
   Executionset( Fullapi )
   Concurrency(Threadsafe)
   Remotesystem()
   Runtime( Le370 )
   Jvmdebug( Nodebug )
    Hotpooling( NOTHotpool )
```

Figure 97. The expanded display of an individual entry (first panel)

```
  IN PROG
  RESULT - OVERTYPE TO MODIFY
   Program(DFH$AALL)
   Jvmdebug( Nodebug )
   Jvmclass(                                           )
   Jvmclass(                                           )
   Jvmclass(                                           )
   Jvmclass(                                           )
   Jvmclass(        )
```

| *Figure 98. The expanded display of an individual entry (second | panel)*

Displayed fields

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.

| **C**oncurrency
| Displays the concurrency attribute of the specified
| program. The values are:

| **T**hreadsafe
| 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.

| **Q**uasirent
| 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.

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

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.

| **D**ynamstatus
| displays whether, if the program is the subject of a
| program link request, the request can be dynamically
| routed. The values are:

| **DY**namic
| 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.

| **NOTDY**namic
| 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.

Hotpooling
displays whether the Java program object is to be run in
a preinitialized Language Environment enclave. The
values are:

HOtpool
The Java program object is to be run in a
preinitialized Language Environment enclave.

NOTHotpool
The Java program object is not to be run in a
preinitialized Language Environment enclave.

| **JVMCL**ass
| displays the name of the class specified on the program
| definition, up to a maximun of 255 characters.

| **JVMDE**bug
| indicates whether the JVM is to operate in debugging
| mode for this program. The values are:

| **N**odebug
| The JVM is not to operate in debugging mode for
| this program.

| **D**ebug
| The JVM is to operate in debugging mode for this
| program.

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

PLi, **CO**bol, **C, LE370, or AS**sembler
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.

| **Note:** It is intented in a future release to report
| C++ programs separately, and for Java
| program objects to return a Language
| value of JAVA. JVM and Java program
| objects will be distinguished by using the
| RUNTIME option, which will report JVM for
| JVM programs and LE370 for Java
| program objects. (A Java program object is
| a Java program that has been bound by
| the VisualAge for Java, Enterprise ToolKit
| for OS/390.)

| **J**ava The program is written in JAVA. This value is
| currently returned for JVM programs only.

| **NOTAP**plic
| 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 0000000 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.

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 that this is not necessarily the same as the number
of times the module has been loaded from dfhrpl.

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

| **J**vm The program in a JAVA program running under
| the control of a JVM.

| **LE370**
| The program is running with LE370 runtime
| support.

NATIVE
The program is running with language-specific
runtime support.

| **UN**known
| The runtime environment in unknown because the
| program has not been loaded by CICS, therefore
| the source language cannot be deduced, which
| dictates the runtime environment to be used.

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

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.

Sample screen

```
  I RE
 STATUS:  RESULT
  Req(DFHGFACT) Omgm(com::ibm::CosLifeCycle )
    Omgi(GenericFactory          )


                                      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 99. 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
    Requestmodel(DFHGFACT)
    Omgmodule(com::ibm::CosLifeCycle)
    Omginterface(GenericFactory)
    Omgoperation(*)
    Transid(CIOF)
```

Figure 100. The expanded display of an individual entry

```
►►──CEMT Inquire REquesmodel──►◄
```

Displayed fields

REQUESTMODEL(*data-value***)**

 specifies an 8-character request model name.

OMGMODULE*(data-area)*

 returns the 58-character field containing the pattern defining the name scope of the interface and operation values for this REQUESTMODEL.

OMGINTERFACE*(data-area)*

 returns the 31-character field containing the pattern matching the interface name for this REQUESTMODEL.

OMGOPERATION*(data-area)*

 returns the 31-character field containing the pattern matching the IDL operation name for this REQUESTMODEL.

| **TRANSID**(name)
| returns the 4-character name of the CICS transaction to
| be executed when a request matching the specification
| of the REQUESTMODEL is received.

| # 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`).
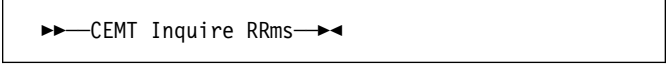
| You will get a display screen, a sample of which is shown
| below.

| **Sample screen**

```
 I RR
 STATUS:  RESULTS
  Rrm Ope
```

| *Figure 101. CEMT INQUIRE RRMS screen*

|
| ►►──CEMT Inquire RRms──►◄

| **Displayed fields**

| **C**losed
| indicates that CICS will not accept inbound transactional
| EXCI work.

| **O**pen
| 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 102. 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 103.

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

Figure 103. The expanded display

```
 ►►──CEMT Inquire STAtistics──►◄
```

Displayed fields

Endofday**(***hhmmss***)**
displays the end-of-day time for recording statistics. At end-of-day, the current values of the CICS statistics counters are written out to an SMF data set and the statistics counters are reset.

End-of-day times must be in the range 000000-235959. The default end-of-day time is 12 midnight (000000).

Interval**(***hhmmss***)**
displays the length of time during which the statistics counters are incremented. At the end of each interval, the accumulated statistics are recorded and the statistics counters are reset.

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

Interval times must be in the range 000100-240000. The default interval value is 030000. The interval value can be adjusted, but will have no effect unless the statistics recording status is set ON. You can set the statistics recording status ON or OFF using the RECORDING option on this command or by using the STATRCD system initialization parameter.

The arrival of the ENDOFDAY time always causes the current interval to be ended (possibly prematurely) and a new interval to be started. Only end-of-day statistics are recorded at the end-of-day time, even if it coincides exactly with the expiry of an interval.hanging 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.

```
  ┌─ CICS initialized with
  │    ENDOFDAY(170000)
  │    INTERVAL(030000)
  │
  │
  │
  │
  │
  ▼
  '─┬┬┬┬─┬┬┬┬─┬┬┬┬─┬┬┬┬─┬┬┬┬─┬┬┬┬─┬┬┬┬─┬┬┬┬─┬┬┬┬─┬┬┬┬
   08   09   10   11   12   13   14   15   16   17
                  I                I              E
```

CEMT INQUIRE STATISTICS

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.

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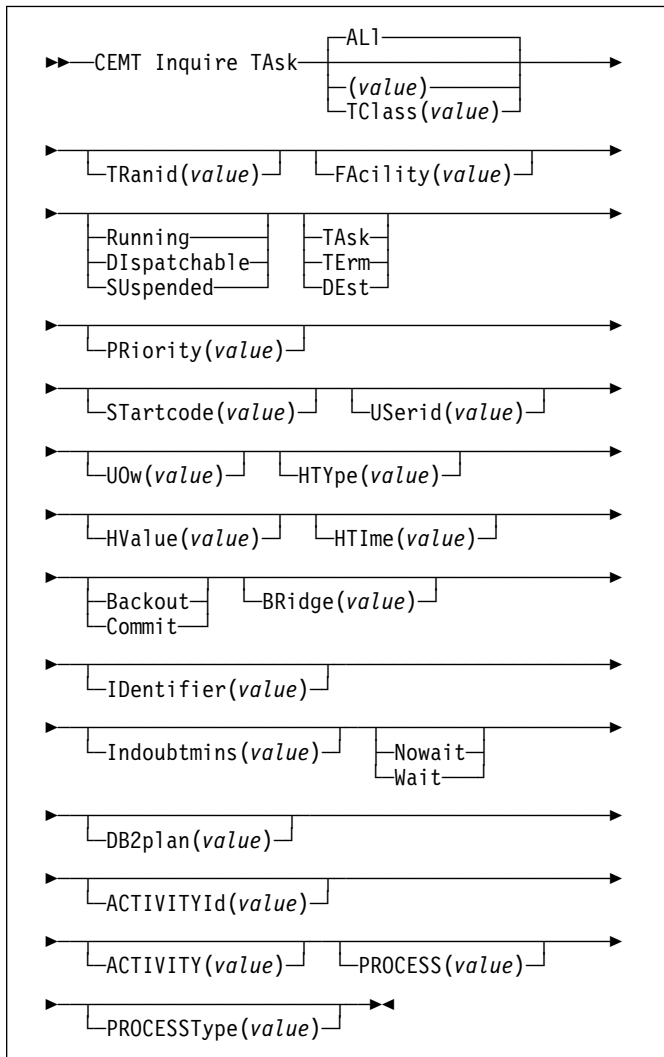
Input: Press the Clear key to clear the screen. There are two ways of commencing this transaction:

- 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 overtype them with the required values.

```
                                              ┌─AL1─┐
  ►►──CEMT Inquire STReamname──────────────────────────────►
  ►────(value)─────────────────────────────────────────────►
                     └─Usecount(value)─┘
  ►──────┬─Failed─┬──┬─Syslog───┬──►◄
         └─Ok─────┘  └─Nosyslog─┘
```
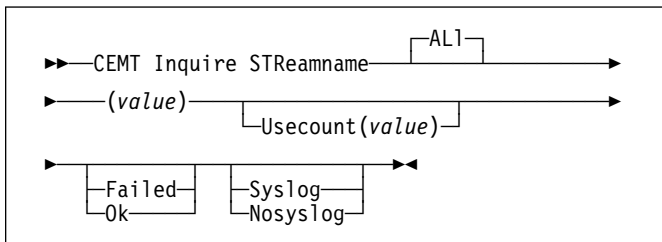
(value**)**
    specifies an MVS system logger log stream name. The name can be a specific LSN of up to 26 characters in length.

ALl
    is the default.

Sample screen

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

Figure 104. 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 105.

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

Figure 105. The expanded display of an individual entry

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 transaction dump command SET TRDUMPCODE can also cause CICS to produce system dumps for the specified transaction dump code.

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

- Type `CEMT INQUIRE SYDUMPCODE` (the minimum abbreviation is `CEMT I SYD`). You get a display that lists the current status.

- Type `CEMT INQUIRE SYDUMPCODE` (`CEMT I SYD`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i syd sy sh`, the resulting display will show you the details of only those system dump codes for which a system dump is to be taken and CICS is to be shut down.

You can then tab to the highlighted or blank fields and overtype them with the required values.
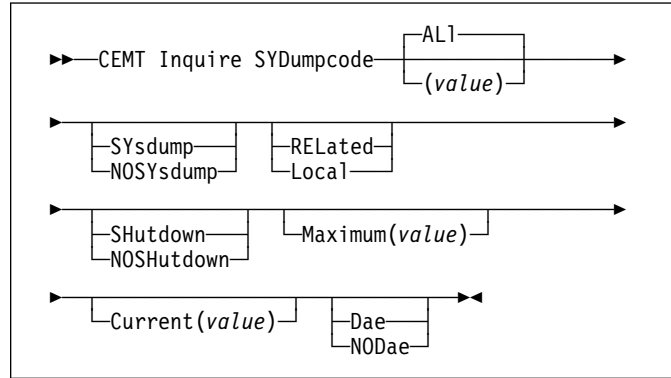
(value**)**
    is an 8-character system dump code. See the *CICS Messages and Codes* manual for details of the codes.

ALl
    is the default.

Sample screen

```
 I SYD
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Syd(AP0001 ) Sys Loc    Max( 010 ) Cur(0000)     Dae
  Syd(AP0002 ) Sys Loc    Max( 999 ) Cur(0000)
```

Figure 106. CEMT INQUIRE SYDUMPCODE screen

Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

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

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

Figure 107. The expanded display

Displayed fields

Action (*input only field*)
specifies the action to be taken with this system dump code. The values are:

Reset
The current number of calls for this dump code is reset to zero.

Remove
This system dump code is removed from the system dump code table.

Add This system dump code is added to the system dump code table.

Current(*value*)
displays the number of dump calls that have been made for this dump code since it was last reset.

Daeoption
displays whether a dump taken for CICS is eligible for suppression by the MVS dump analysis and elimination (DAE) component. The values are:

Dae A system dump is eligible for suppression by the DAE component.

NODae
A system dump will not be suppressed by the DAE component. (However, be aware of the SUPPRESS and SUPPRESSALL options in the ADYSETxx parmlib member. These are controlled by the VRADAE and VRANODAE keys in the SDWA. They may lead to dump suppression even though NODAE is set here. For information about DAE, SUPPRESS, and SUPPRESSALL, see the *OS/390 MVS Diagnosis: Tools and Service Aids* manual.

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

Dumpscope(*value*)
displays whether a system dump request is to be sent to MVS images in the sysplex which are running XCF/MRO connected CICS regions related to the CICS region on which the dump is initiated.

Note: A related CICS region is one in which the unit of work identifiers, in the form of APPC tokens, of one or more tasks match those in the CICS region that issued the dump request.

RELated
A system dump request is to be sent for this system dump code.

Local
A system dump request is not to be sent to other MVS images in the sysplex for this system dump code.

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

Maximum(*value*)
displays the maximum number of dump calls for this system dump code which result in system dumps being taken. The value is in the range 0-999 and can be specified by using SYDUMAX in the SIT parameter. A value of 999 means the default, 'no limit'.

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

Shutoption
displays whether CICS is to be shut down after the occurrence of an error corresponding to this dump code.

SHutdown
CICS is to be shut down.

NOSHutdown
CICS is not to be shut down.

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

SYDumpcode(*value*)
indicates that this panel relates to a SYDUMPCODE inquiry and displays an 8-character system dump code. See 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 overtype them with the required values.

```
►►──CEMT Inquire SYStem──►◄
```

To change various attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see Figure 108)
- Use the CEMT SET SYSTEM command.

Sample screen

```
 inq sys
 STATUS:  RESULTS - OVERTYPE TO MODIFY
   Actopentcbs(000)              Progautoctlg( Ctlgmodify )
   Aging( 32768 )                Progautoexit( DFHPGADX )
   Akp( 04000 )                  Progautoinst( Autoactive )
   Cicstslevel(010300)           Reentprotect(Reentprot)
   Cmdprotect(Nocmdprot)         Release(0530)
   Db2conn()                     Runaway( 0005000 )
   Dfltuser(BELL)                Scandelay( 0500 )
   Dsalimit( 05242880 )          Sdtran(CESD)
   Dsrtprogram( NONE    )        Sosstatus(Notsos)
   Dtrprogram( DFHDYP   )        Storeprotect(Active)
   Dumping( Sysdump )            Time( 0001000 )
   Edsalimit( 0020971520 )       Tranisolate(Active)
   Forceqr( Noforce )
   Logdefer( 00030 )
   Maxopentcbs( 005 )
   Maxtasks( 032 )
   Mrobatch( 001 )
   Oslevel(020500)

                                  SYSID=HT61 APPLID=CICSHT61
   RESPONSE: NORMAL               TIME: 11.10.50  DATE: 11.04.98
 PF 1 HELP      3 END     5 VAR  7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 108. CEMT INQUIRE SYSTEM screen

Displayed fields

Actopentcbs**(**value**)**
   displays the number of open TCBs of all modes 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.

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 *vvrrmm*, and CICS TS Release 3 returns 010300.

CMdprotect**(**value**)**
   displays whether command protection, which validates start addresses passed on in CICS commands, is active or not (that is, whether the CMDPROT system initialization parameter specifies YES or NO). The values are:

   CMdprot
      Command protection is active. CICS checks to ensure that the task itself has write access to the storage referenced on the command before writing to the storage on the task's behalf.

   NOCmdprot
      Command protection is not active. CICS does not check to ensure that the task itself has write access to the storage referenced on the command before writing to the storage on the task's behalf.

Db2conn**(**value**)**
   displays the name of the currently installed DB2CONN.

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.

| **DT**rprogram(*pgrmid*)
| displays the name of the dynamic routing program.

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

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.

| **F**orceqr(*value*)
| displays whether CICS is to force user application
| programs that are defined as threadsafe to run on the
| quasi-reentrant (QR) TCB. The values are:

| **F**orce
| 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).

| **N**force
| 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.

| **L**ogdefer(*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.

| **M**axopentcbs(*value*)
| displays the maximum number of open TCBs, of all
| modes, that CICS is allowed to attach. The value can be
| in the range 1–999.

| **Note:** Until open TCB support is extended to allow
| task-related user exits to run on CICS-managed
| TCBs (mode L8), this value is applicable to JVM
| TCBs only (mode J8).

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 0240.

Progautoctlg
displays whether autoinstalled program definitions are to be cataloged. The values are:

CTLGAll
All autoinstalled program definitions are to be cataloged and restored on a warm or emergency start.

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 start.

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

Progautoexit(*value*)
displays the name of the user-provided program that is called by the program autoinstall code to select or modify a model definition.

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

Progautoinst
displays whether autoinstall for programs is active or inactive. The values are:

AUTOActive

Autoinstall for programs is active. On first use, if a program, mapset, or partitionset is not defined, the definition is created dynamically.

AUTOInactive

Autoinstall is not active. If a program is not defined, a PGMIDERR or transaction abend occurs when it is referenced.

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

Reentprotect**(**value**)**

displays whether read-only storage is in use for reentrant programs (that is, whether the RENTPGM system initialization parameter specifies PROTECT or NOPROTECT). The values are:

NOReentprotect

CICS allocates storage for the read-only DSAs (RDSA and ERDSA) from CICS-key storage. Reentrant programs do not have the protection of residing in read-only storage, and can be modified by programs executing in CICS key.

REEntprotect

CICS allocates storage for the RDSA and ERDSA from key-0, non-fetch protected, storage. CICS loads reentrant programs into this storage, and the programs are protected by residing in read-only storage.

RELease**(**value**)** (supported for compatibility only)

displays a 4-digit number (value 0530 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.

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

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

NOTsos

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

SOSAbove

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

SOSBelow

CICS is not short on storage in any of the dynamic storage areas below 16MB.

SOS 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:

ACtive

CICS is operating with the CICS storage protection facility.

Inactive

CICS is operating without the CICS storage protection facility.

See the description of the 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:

ACtive

Transaction isolation is active in the CICS region.

Inactive

CICS is running without transaction isolation, either because the support is not available, or it was not requested at CICS initialization.

CEMT INQUIRE TASK

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:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see "Overtyping a display" on page 49).

- Use the CEMT SET TASK command.

```
                              ┌─ALl──┐
 ▶▶──CEMT Inquire TAsk────────┼──────┼────────────────────────▶
                              ├─(value)──┤
                              └─TClass(value)─┘

 ▶─────┬──────────────────┬────┬──────────────────┬────────────▶
       └─TRanid(value)─┘        └─FAcility(value)─┘

 ▶─────┬─────────────────┬──────┬────────┬────────────────────▶
       ├─Running──────┤          ├─TAsk─┤
       ├─DIspatchable─┤          ├─TErm─┤
       └─SUspended────┘          └─DEst─┘

 ▶─────┬──────────────────┬──────────────────────────────────▶
       └─PRiority(value)─┘

 ▶─────┬───────────────────┬────┬─────────────────┬──────────▶
       └─STartcode(value)─┘      └─USerid(value)─┘

 ▶─────┬──────────────┬──────┬───────────────┬───────────────▶
       └─UOw(value)─┘        └─HTYpe(value)─┘

 ▶─────┬─────────────────┬──────┬───────────────┬────────────▶
       └─HValue(value)─┘        └─HTIme(value)─┘

 ▶─────┬─────────────┬──────┬────────────────┬───────────────▶
       ├─Backout─┤            └─BRidge(value)─┘
       └─Commit──┘

 ▶─────┬─────────────────────┬───────────────────────────────▶
       └─IDentifier(value)─┘

 ▶─────┬───────────────────────┬──────┬────────┬─────────────▶
       └─Indoubtmins(value)─┘          ├─Nowait─┤
                                       └─Wait───┘

 ▶─────┬────────────────┬────────────────────────────────────▶
       └─DB2plan(value)─┘

 ▶─────┬────────────────────┬────────────────────────────────▶
       └─ACTIVITYId(value)─┘

 ▶─────┬──────────────────┬──────┬─────────────────┬─────────▶
       └─ACTIVITY(value)─┘        └─PROCESS(value)─┘

 ▶─────┬──────────────────────┬───────────────────────────────▶◀
       └─PROCESSType(value)─┘
```

(value**)**
>   is the CICS-generated task number, in the range 1–99999.

ALl
>   is the default.

TClass**(**value**)**
>   is the 8-character transaction class name to which the transaction belongs.

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 109. CEMT INQUIRE TASK screen

Note: There are blank fields on the screen where a value does not apply or is 'negative'; that is, the value begins with 'No'. To modify these fields, locate them by tabbing (they appear in the same sequence as in the expanded format), and overtype with input valid for that field. You may find it more convenient to use the expanded format when setting one of these values, (negating is easier because you are overtyping a displayed field with its negative causing the result to become non-displayed).

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 110.

```
 I TASK
 RESULT - OVERTYPE TO MODIFY
  Task(0000026)
  Tranid(CEMT)
  Facility(S262)
  Runstatus(Suspended)
  Ftype(Term)
  Priority( 255 )
  Purgetype(           )
  Startcode(TO)
  Userid(CICSUSER)
  Uow(B0C9D8D111440007)
  Htype(ZCIOWAIT)
  Hvalue(DFHZARQ1)
  Htime(000159)
  Indoubt(Backout)
  Indoubtwait(Wait)
  Bridge()
  Identifier()
  Indoubtmins(000000)
  Db2plan()
  Activityid()
  Activity()
  Process()
  Processtype()
  Tcb(QR)



                               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 110. The expanded display of an individual entry

Displayed fields

Activity(value)
>   displays the 16-character, user-assigned, name of the CICS business transaction services activity that this task is executing on behalf of.

Activityid(value)
>   displays the 52-character, CICS-assigned, identifier of the CICS business transaction services activity that this task is executing on behalf of.

BRidge**(**value**)**
>   Returns the 4-character name of the bridge monitor transaction if the current task is running in a 3270 bridge environment, and was started by a bridge monitor transaction with a START BREXIT TRANSID command. Otherwise, blanks are returned.

FAcility**(**value**)**
>   displays a 4-character string identifying the name of the terminal or queue that initiated the task. If no FACILITY value is displayed, the task was started without a facility.

Ftype
>   displays the type of facility that initiated this task. The values are:

TAsk

The task was initiated from another task.

TErm

The task was initiated from a terminal.

DEst The task was initiated by a destination trigger level as defined in the destination control table (DCT).

HTIme**(***value***)**

displays the time (in seconds) that the task has been in the current suspended state.

HTYpe**(***value***)**

displays the reason why the task is suspended. A null value indicates that there is no hold-up, except for the necessity of reaching the head of the queue.

HValue**(***value***)**

| displays a 16-character resource name, such as a file
| name, or a value such as a TCLASS value.

For information on the values that can appear in the HTYPE and HVALUE options, and how they can be used as an aid in problem determination, see the "resource type" and "resource name" details in the *CICS Problem Determination Guide*.

Indoubtmins**(***value***)**

displays the length of time, in minutes, after a failure during the in-doubt period, before the task is to take the action returned in the Indoubt field. The returned value is valid only if the unit of work is in-doubt and Indoubtwait returns WAIT.

See also Indoubt and Indoubtwait.

Indoubt**(***value***)**

displays the action (based on the ACTION attribute of the TRANSACTION resource definition) to be taken if the CICS region fails or loses connectivity with its coordinator while a unit of work is in the in-doubt period.

The action is dependent on the values returned in Indoubtwait and Indoubtmins; if Indoubtwait returns WAIT, the action is not taken until the time returned in Indoubtwait expires.

The values are:

Backout

All changes made to recoverable resources are to be backed out.

Commit

All changes made to recoverable resources are to be committed, and the unit of work marked as completed.

Indoubtwait**(***value***)**

displays how (based on the WAIT attribute of the TRANSACTION definition) a unit of work (UOW) is to respond if a failure occurs while it is in an in-doubt state. The values are:

Nowait

The UOW is not to wait, pending recovery from the failure. CICS is to take immediately whatever action is specified on the ACTION attribute of the TRANSACTION definition.

Wait The UOW is to wait, pending recovery from the failure, to determine whether recoverable resources are to be backed out or committed.

For further information about the meaning of the ACTION and WAIT attributes of the TRANSACTION definition, see the *CICS Resource Definition Guide*.

IDentifier**(***value***)**

| returns a 48-character field containing user data
| provided by the bridge exit, if the task was initiated in
| the 3270 bridge environment, or blanks, otherwise. This
| field is intended to assist in online problem resolution.

| For example, it could contain the 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.

| **Process**type*(value)*

| displays the 8-character process-type of the CICS
| business transaction services process that this task is
| executing on behalf of.

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.

Purgetype **(***input only field***)**

specifies whether a task is to be purged or forcepurged. The values are:

Purge

The task is to be terminated. Termination occurs only when system and data integrity can be maintained.

Forcepurge

The task is to be terminated immediately. System integrity is not guaranteed. In some extreme cases, for example if a task is forcepurged during backout processing, CICS terminates abnormally. If you want to terminate a task but do not want to terminate CICS, you should use PURGE instead of FORCEPURGE.

In some BTAM situations, further user action is required to complete the purging of the task. If a task is in a terminal read on a BTAM nonlocal

terminal and that task is canceled, it can require input from another terminal on the same line before the cancel operation completes.

Runstatus
  displays the status of this task. The values are:

  Running
     The task is running.

  DIspatchable
     The task is dispatchable.

  SUspended
     The task is suspended.

STartcode**(***value***)**
  displays how this task was started. The values are:

  D A distributed program link (DPL) request. The program cannot issue I/O requests against its principal facility or any syncpoint requests.

  DS A distributed program link (DPL) request, as for code D, with the exception that the program can issue syncpoint requests.

  QD A transient data trigger level was reached.

  S Start command (no data)

  SD Start command (with data)

  TO The operator typed a transaction code at the terminal.

  TP The transaction was started by presetting the transaction ID for the terminal.

  U User-attached task.

TAsk**(***value***)**
  indicates that this panel relates to a TASK inquiry and displays a CICS-generated task number in the range 1–99999.

| **T**cb**(***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 JVM (mode J8) TCB).

| **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 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.

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.
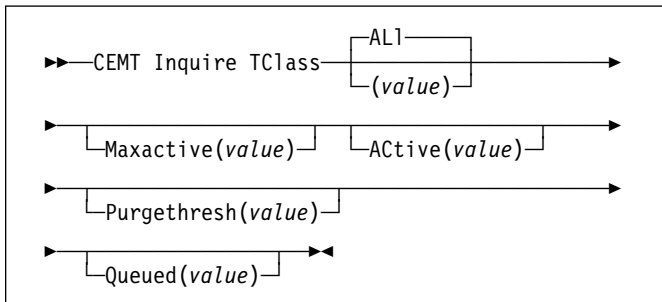
(value**)**
   is the 8-character transaction class name.

ALl
   is the default.

Sample screen

```
  IN TCLASS
  STATUS:  RESULTS - OVERTYPE TO MODIFY
    Tcl(DFHTCIND) Max( 010 ) Act(000) Pur( 0000000 )
       Que(000000)
    Tcl(DFHTCL01) Max( 001 ) Act(000) Pur( 0000000 )
       Que(000000)
    Tcl(DFHTCL02) Max( 001 ) Act(000) Pur( 0000000 )
       Que(000000)
    Tcl(DFHTCL03) Max( 001 ) Act(000) Pur( 0000000 )
       Que(000000)
    Tcl(DFHTCL04) Max( 001 ) Act(000) Pur( 0000000 )
       Que(000000)
    Tcl(DFHTCL05) Max( 001 ) Act(000) Pur( 0000000 )
       Que(000000)
    Tcl(DFHTCL06) Max( 001 ) Act(000) Pur( 0000000 )
       Que(000000)
    Tcl(DFHTCL07) Max( 001 ) Act(000) Pur( 0000000 )
       Que(000000)
  + Tcl(DFHTCL08) Max( 001 ) Act(000) Pur( 0000000 )
       Que(000000)
```

Figure 111. 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 112.

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

Figure 112. 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:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see "Overtyping a display" on page 49).
- Use the CEMT SET TCPIP command.

```
►►──INQUIRE TCPIP──────────────────────►◄
                    └─OPENSTATUS(value)─┘
```
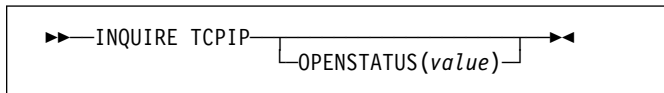
OPENSTATUS(value)
displays the status of CICS internal TCP/IP support. The values are:

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.

CEMT INQUIRE TCPIPSERVICE

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 TCPIPSERVICE returns information about the state of a service using CICS internal TCP/IP support.

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

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

To change various attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see "Overtyping a display" on page 49).
- Use the CEMT SET TCPIPSERVICE command.

```
►►──INQUIRE TCPIPSERVICE(value)─────────────────►
►──────────────────────────────────────────────►
   └─Authenticate(cvda)─┘
►──────────────────────────────────────────────►
   └─Backlog(value)─┘
►──────────────────────────────────────────────►
   └─CLOsetimeout(value)─┘
►──────────────────────────────────────────────►
   └─Connection(value)─┘
►──────────────────────────────────────────────►
   └─IPaddress(value)─┘
►──────────────────────────────────────────────►
   ┌─OPENSTATUS(value)─┐  ┌─Port(value)─┐
   ├─OPEN──────────────┤  └─────────────┘
   ├─CLOSING───────────┤
   ├─CLOSED────────────┤
   └─IMMCLOSING────────┘
►──────────────────────────────────────────────►
   ┌─SSLTYPE(value)─┐
   ├─NO─────────────┤
   ├─YES────────────┤
   └─CLIENTAUTH─────┘
►──────────────────────────────────────────────►
   ┌─SOcketclose(value)─┐
   ├─Wait───────────────┤
   └─TImeout────────────┘
►──────────────────────────────────────────────►
   └─TCPIPService(value)─┘
►──────────────────────────────────────────────►
   └─TRansid(value)─┘  └─TSqprefix(value)─┘
►─────────►◄
   └─URM(value)─┘
```
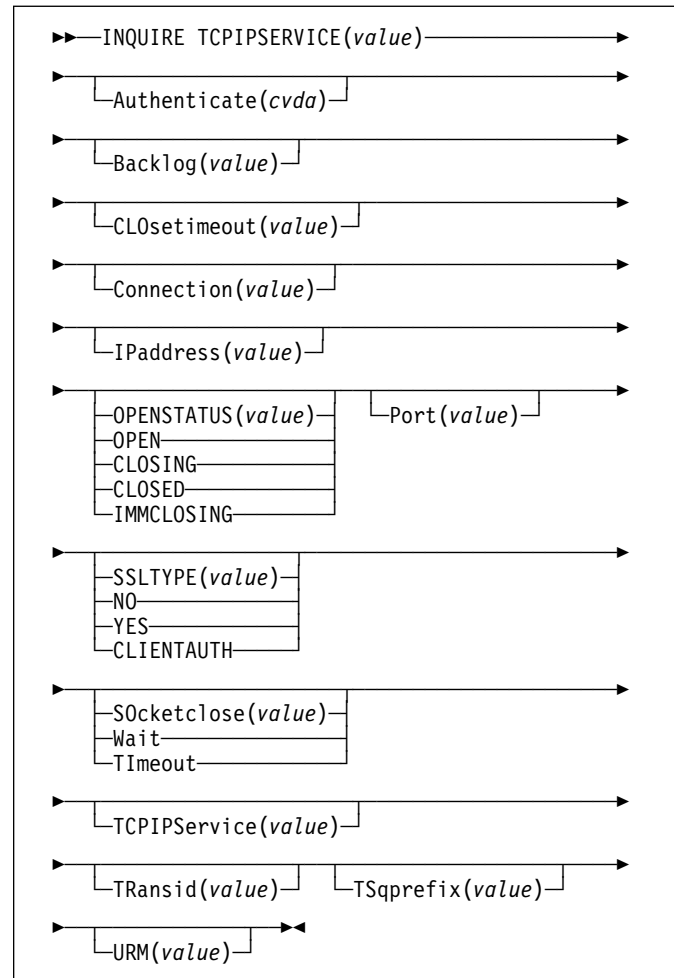
Authenticate

returns a CVDA value indicating the authentication
option for this service. CVDA values are:

NOAUTHENTIC
No authentication of the client is forced, but if a
registered certificate is received it is used to
determine a userid.

BASIC
Basic authentication of the client is attempted.

CERTIFICATE
A valid trusted, registered, certificate is required.

REGISTER
A client certificate is required. If it is not registered
to a userid, BASIC authentication is attempted and
the certificate is registered to the resulting userid.

AUTOMATIC
Either CERTIFICATE or BASIC authentication is
used.

| **Backlog**
| Change the maximum number of requests which can be
| queued in TCP/IP waiting to be processed by the
| service.

| **Closetimeout**
| returns a fullword value containing the number of
| seconds specified for the timeout period. This can be 0
| through 86400 (24 hours).

| **Connection**
| The number of current sockets connections for this
| service

| **Ipaddress**
| returns an IP address. If you specify an IP address in
| the TCPIPSERVICE definition, that address is returned,
| otherwise the default IP address is returned. If there is
| more than one IP address on an OS/390 system, only
| the default is returned.

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

| **Service**
| 8-character name identifying this service.

| **Socketclose**
| returns a CVDA value indicating whether a timeout value
| is in effect for the TCPIPSERVICE. CVDA values are:

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

| **TIMEOUT**
| A value was specified for the SOCKETCLOSE
| parameter on the definition.

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

| **YES**
| Non client authentication SSL flows are processed
| on this port

| **NO**
| Non SSL flows are processed on this port

| **CLIENTAUTH**
| SSL client authentication flows are processed on
| this port

| **STATUS(cvda)**
| returns a CVDA value indicating the state of the service.
| CVDA 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.

| **Transid**
| 4-character transaction id used on the attach for the task
| started to process a new request.

| **TSQprefix**
| returns the 6–character prefix used when the service
| generates TS queue names.

| **URM**
| 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 the destination control table (DCT).

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

- Type `CEMT INQUIRE TDQUEUE` (the minimum abbreviation is `CEMT I TD`). You get a display that lists the current status.

- Type `CEMT INQUIRE TDQUEUE` (`CEMT I TD`) followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i td ind en`, the resulting display will show you the details of only those transient data queues that are indirect and enabled.

To change various attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see "Overtyping a display" on page 49).

- Use the CEMT SET TDQUEUE command.
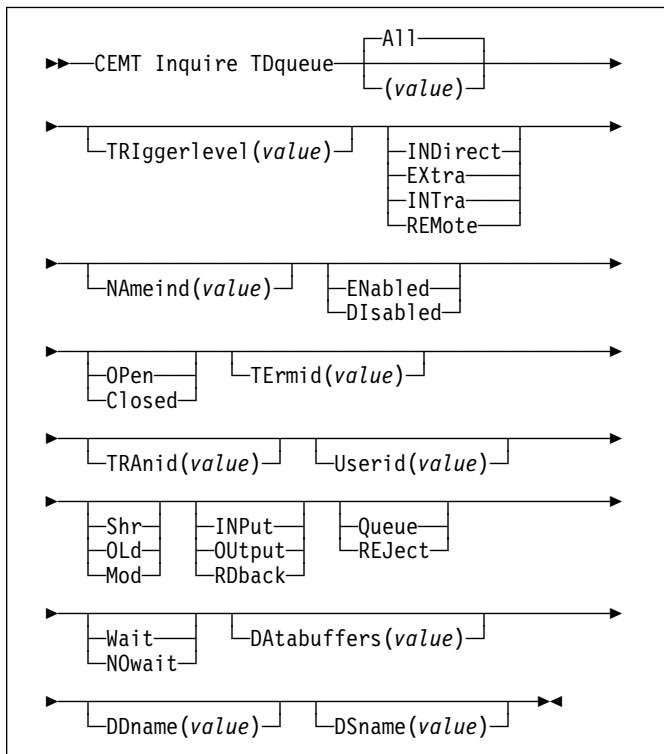
(value**)**
    is the identifier (1-4 characters) of a transient data queue.

All is the default.

Sample screen

```
 IN TDQ
  STATUS:  RESULTS - OVERTYPE TO MODIFY
   Tdq(CADL)            Ind Nam(CSSL)
   Tdq(CAIL)            Ind Nam(CSSL)
   Tdq(CCPI)            Ind Nam(CSSL)
   Tdq(CCSE)            Ind Nam(CCSO)
   Tdq(CCSO)            Ext         Ena Ope
                        Mod Out Dat(001) Ddn(COUT   )
   Tdq(CDBC)            Ind Nam(CSSL)
   Tdq(CDUL)            Ind Nam(CSSL)
   Tdq(CESE)            Ext         Ena Ope
                        Shr Out Dat(001) Ddn(CEEMSG )
   Tdq(CESO)            Ext         Ena Ope
                        Shr Out Dat(001) Ddn(CEEOUT )
   Tdq(CMIG)            Ind Nam(CSSL)
   Tdq(CPLD)            Ind Nam(CPLI)
   Tdq(CPLI)            Ext         Ena Ope
                        Mod Out Dat(001) Ddn(PLIMSG )
   Tdq(CRDI)            Ind Nam(CSSL)
 + Tdq(CSCS)            Ind Nam(CSSL)
```

Figure 113. 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 114.

```
 IN TDQ
  RESULT - OVERTYPE TO MODIFY
   Tdqueue(CCSO)
   Triggerlevel(      )
   Type(Extra)
   Nameind()
   Enablestatus( Enabled )
   Openstatus( Open )
   Termid()
   Tranid()
   Userid()
   Disposition(Mod)
   Iotype(Output)
   Indoubt()
   Indoubtwait()
   Databuffers(001)
   Ddname(COUT)
   Dsname(BELL.CICSHT61.JOB04382.D0000105.?)
```

Figure 114. The expanded display of an individual entry

Displayed fields

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.

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).

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.

| **M**ember **(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.

NAmeind**(***value***) (indirect queues only)**
displays a 4-character string identifying the name of the queue pointed to by the indirect queue.

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.

Queue
The UOW is in-doubt and waiting; any locks held by the UOW for this queue remain active until the final state of the UOW is known. This means that tasks are suspended rather than receiving the LOCKED response. When the final state of the UOW is known, any changes that it has made are committed or backed out. Until then, any further requests of the following types that need one of the active locks must wait:

- READQ, if the indoubt UOW has issued READQ or DELETEQ requests

- WRITEQ, if the indoubt UOW has issued WRITEQ or DELETEQ requests

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

REJect
The UOW is in-doubt and is waiting. Any locks held by the UOW for this queue are retained until the final state of the UOW is known. When the final state is known, any changes the UOW has made are committed or backed out. Until then, any further requests that need one of the retained locks are rejected, and a LOCKED response is returned. INDOUBT=REJECT causes a LOCKED response to be raised in exactly the same circumstances as those in which INDOUBT=QUEUE causes a transaction to wait.

Tdqueue**(***value***)**
indicates that this panel relates to a TDQUEUE inquiry and displays the 4-character identifier of a transient data queue. Queue names beginning with "C" are normally reserved for use by CICS.

TErmid**(***value***)**
displays the 4-character name of the terminal or session to be associated with this queue when automatic transaction initiation (ATI) occurs. See also **Tra**nid and **Tri**ggerlevel.

TRAnid**(***value***)**
displays the 4-character identifier of the transaction that is to be initiated automatically when the queue trigger level is reached.

TRIggerlevel**(***value***) (intrapartition queues only)**
displays the number of requests for output to a queue that must accrue before automatic transaction initiation (ATI) occurs.

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

The number can be between 0 and 32767.

Type
displays the type of this transient data queue. The values are:

INDirect
The queue type is indirect. The name of the final target queue is shown in the next field.

EXtra
The queue type is extrapartition.

INTra
The queue type is intrapartition.

REMote
The queue type is remote.

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

Indoubtwait**(***value***) (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.

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.

DAtabuffers**(***value***) (extrapartition queues only***)***
displays the number of buffers that can be used by the associated QSAM data set in the range 1 through 255.

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

DDname**(***value***) (extrapartition queues only***)***
displays a 1-8 character value that may refer to a data set defined in the startup JCL.

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.

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 or ACF/TCAM GSA 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 100.

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 cemt i te i at, the resulting display will show you the details of only those terminals that are in service and available for use. Or if you enter cemt 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:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see "Overtyping a display" on page 49).

- Use the CEMT SET TERMINAL command.
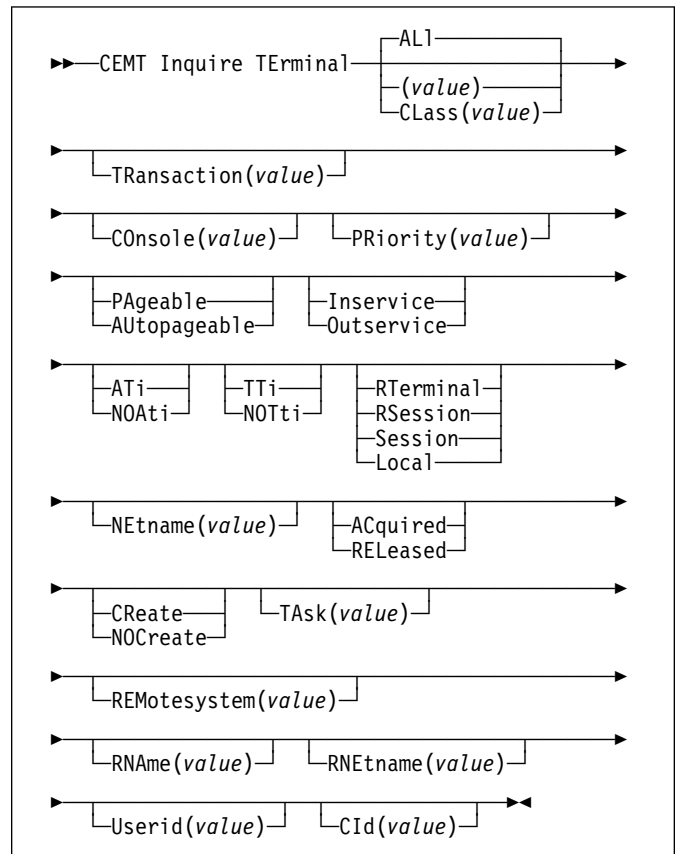
(_value_**)**

is a terminal identifier (1-4 characters) as specified in an installed terminal definition.

If the terminal name is, for example, S201, this option is coded thus:

CEMT INQUIRE TERMINAL(S201)

ALl

is the default. For example, if you inquire about terminals, you receive information about all terminals, unless you specify a terminal identifier, a terminal class identifier, or a system identifier.

CLass**(**_value_**)**

is the 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.

CEMT INQUIRE TERMINAL

Sample screen

```
IN TERM
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Ter(CBRF)         Pri( 000 ) Pag Ins Ati Tti Rte
   Net(CBRF )                    Rem(CBR ) Rna(CBRF)
 Ter(CERR)         Pri( 000 ) Pag Ins    Tti Loc
   Con(*ERRCON* )
 Ter(CTSO)         Pri( 000 ) Pag Ins    Tti Loc
   Con(PKING     )
 Ter(KING)         Pri( 000 ) Pag Ins    Tti Loc
   Con(PRUEKING  )
 Ter(RNAL)         Pri( 000 ) Pag Ins    Tti Loc
   Con(INTERNAL  )
 Ter(S225) Tra(CEMT) Pri( 000 ) Pag Ins Ati Tti Loc
   Net(IGCS225 ) Acq    Tas(0000068)
 Ter(S280) Tra(CEMT) Pri( 000 ) Pag Ins Ati Tti Loc
   Net(IGCS280 ) Acq    Tas(0000063)
 Ter(TC12) Tra(CEMT) Pri( 000 ) Pag Ins Ati Tti Loc
   Net(IYCWTC12) Acq    Tas(0000072)
 Ter(-AAZ)         Pri( 000 ) Aut Ins Ati Tti Ses
   Net(TMPLATE1) Rel            Rem(CBPS)
 Ter(-AA0)         Pri( 000 ) Aut Ins Ati Tti Ses
   Net(TMPLATE1) Rel Cre        Rem(CBPS)
+ Ter(-AA1)        Pri( 000 ) Aut Out Ati Tti Ses
   Net(TMPLATE1) Rel Cre        Rem(CBPS)
```

Figure 115. 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 116.

```
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()
  Rname()
  Rnetname()
+ Cid()
```

Figure 116. 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 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 rest 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 a remote device.

RSession

> The terminal is a session of a remote connection.

Session

> The terminal is an ISC or IRC session.

Local

> The terminal is directly attached to this system and is a physical device or a function of a cluster controller.

NEtname(*value*)

displays an 8-character network name.

For a physical terminal, the Netname is the name by which this terminal is known to VTAM.

For ISC sessions, it is the name by which the session (or group of sessions) is known to VTAM.

For IRC sessions, it is the name used by the connected region to log on to the interregion communication program (DFHIRP).

For a remote terminal, it is the name by which the terminal is known to the VTAM in the remote region. (For a remote terminal routed from a pre-CICS Transaction Server for OS/390 region, Netname is blank.)

Pagestatus

displays whether pages after the first in a series are written to the terminal on request from the operator or automatically. The values are:

PAgeable

> Pages are written on request.

AUtopageable

> Pages are written automatically.

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

PRiority(*value*)

displays a 3-character string identifying the priority of a terminal relative to other terminals. The priority of a task is the sum of the transaction priority, the terminal priority, and the operator priority. Priority has no meaning for terminals that are ISC sessions being used as alternative facilities.

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

The value is in the range 0–255, where 255 is the highest priority.

Purgetype **(***input only field***)**

specifies whether transactions running with the named terminal are to be purged. The values are:

Purge

> Transactions 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 **Net**name.

Termstatus **(***vtam only***)**

displays whether CICS is in session with the logical unit represented by this terminal. The values are:

ACquired

CICS is in session with the logical unit represented by the terminal.

RELeased

CICS is not in session with the logical unit represented by the terminal. If you set this option to RELEASED, a session is terminated immediately if you also specify the PURGE option, otherwise the session is terminated when the current active transaction finishes.

For physical terminals and LU6.1 sessions, you can reset this value by overtyping it with a different value or with:

Coldacq

CICS is in session with the logical unit represented by the terminal, where no resynchronization is required.

TRansaction**(***value***)**

displays a 4-character string identifying the name of the transaction currently being processed with this terminal as its principal facility or as a secondary facility.

Ttistatus

displays whether the terminal can be used by the transactions that are initiated from this terminal. The values are:

TTi   This terminal can be used by transactions.

NOTti

This terminal cannot be used by transactions.

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

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:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see "Overtyping a display" on page 49).

- Use the CEMT SET TRANSACTION command.

```
►►──CEMT Inquire TRAnsaction──────────────────►

      ┌─ALl─────────┐
      │             │
      ├─(value)─────┤    ┌─PRIority(value)─┐
      └─CLass(value)┘    └─────────────────┘

          ┌─PROgram(value)─┐    ┌─TClass(value)─┐
          └────────────────┘    └───────────────┘

      ┌─Enabled──┐    ┌─PUrgeable─────┐
      └─Disabled─┘    └─NOTPurgeable──┘

          ┌─PRFile(value)─┐    ┌─CDatakey─┐
          └───────────────┘    └─Udatakey─┘

      ┌─ANy───┐    ┌─ISolate──┐
      └─BElow─┘    └─NOIsolate┘

          ┌─TRprof(value)─┐    ┌─BAckout─┐
          └───────────────┘    └─COmmit──┘

          ┌─BRexit(value)─┐
          └───────────────┘

          ┌─FAcilitylike(value)─┐
          └─────────────────────┘

      ┌─INdoubtmins(value)─┐    ┌─NOWait─┐
      └────────────────────┘    └─Wait───┘

      ┌─Routable────┐
      └─NOTRoutable─┘    ►◄
```
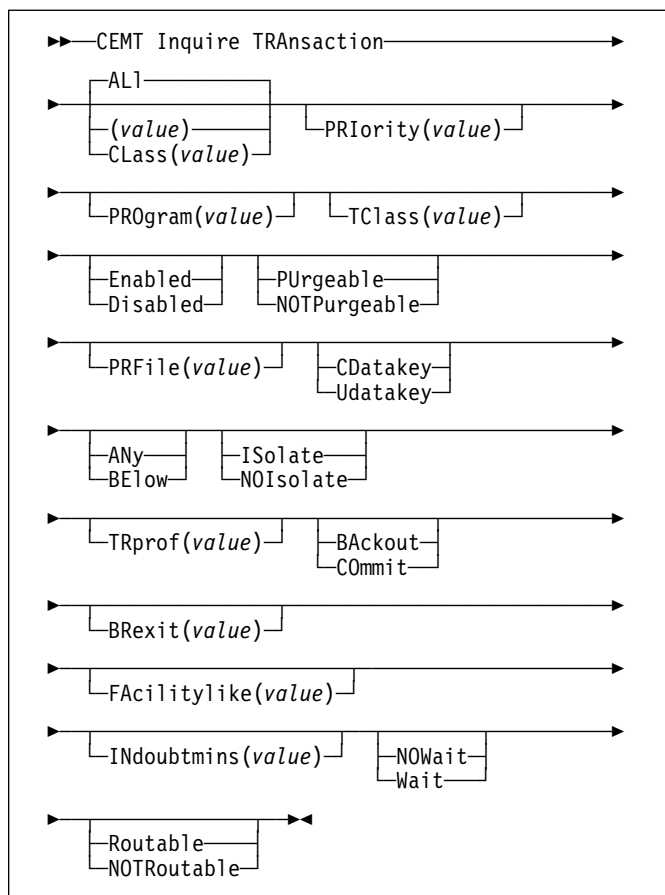
(value)
    is a 1–4 character transaction identifier. Only transactions that have been defined in the CICS system definition (CSD) file and installed on the running CICS system are accessible through CEMT.

ALl
    is the default.

CLass**(value)**
    is the 2-character suffix of a transaction list table (XLT).

Sample screen

```
 IN TRAN
 STATUS:  RESULTS - OVERTYPE TO MODIFY
  Tra(AADD) Pri( 001 ) Pro(DFH$AALL) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
  Tra(ABRW) Pri( 001 ) Pro(DFH$ABRW) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
  Tra(ADYN) Pri( 001 ) Pro(DFH99   ) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
  Tra(AINQ) Pri( 001 ) Pro(DFH$AALL) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
  Tra(AMNU) Pri( 001 ) Pro(DFH$AMNU) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
  Tra(AORD) Pri( 001 ) Pro(DFH$AREN) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
  Tra(AORQ) Pri( 001 ) Pro(DFH$ACOM) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
  Tra(AREP) Pri( 001 ) Pro(DFH$AREP) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
+ Tra(AUPD) Pri( 001 ) Pro(DFH$AALL) Tcl( DFHTCL00 ) Ena Pur
     Prf(DFHCICST) Uda Bel Iso
```

Figure 117. 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 118.

```
 IN TRAN
 RESULT - OVERTYPE TO MODIFY
  Transaction(AADD)
  Priority( 001 )
  Program(DFH$AALL)
  Tclass( DFHTCL00 )
  Status( Enabled )
  Routing(Dynamic)
  Routestatus(Notroutable)
  Purgeability( Purgeable )
  Prfile(DFHCICST)
  Taskdatakey(Udatakey)
  Taskdataloc(Below)
  Isolatest(Isolate)
  Trprof()
  Indoubt
  Brexit
```

Figure 118. The expanded display of an individual entry

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.

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.

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.

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 OS/390 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.

PRFile**(**_value_**)**

displays the name of the profile definition that defines additional options associated with this transaction.

PRIority**(**_value_**)**

displays a value indicating the priority of a transaction relative to other transactions. When a transaction is running as a CICS task, the priority of a task is the sum of the transaction priority, the terminal priority, and the operator priority.

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

The value is in the range 0–255, where 255 is the highest priority.

PROgram**(**_value_**)**

displays an 8-character string identifying the name of the first program to be executed when this transaction is started.

Purgeability

displays whether the transaction is purgeable in system stall conditions. The values are:

PUrgeable

The transaction is system-purgeable. This value relates to the SPURGE parameter on the transaction resource definition and indicates that CICS can purge the transaction in a deadlock time-out situation. See the _CICS Resource Definition Guide_ for information about the SPURGE and DTIMEOUT parameters on a transaction resource definition.

NOTPurgeable

The transaction cannot be purged.

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

Routstatus

displays whether, if the transaction is the subject of an eligible EXEC CICS START command, it will be routed using the enhanced routing method. The values are:

| **NOTR**outable
| If the transaction is the subject of a START
| command, it will be routed using the "traditional"
| method.

| **R**outable
| If the transaction is the subject of an eligible
| START command, it will be routed using the
| enhanced method.

| For details of the enhanced and "traditional" methods of
| routing transactions invoked by EXEC CICS START
| commands, see the *CICS Intercommunication Guide*.

Status
displays whether the transaction is available for use. The
values are:

Enabled
The transaction is available for use.

Disabled
The transaction is not available for use.

Note: If a transaction is disabled, this does not
prevent a START command that names
this transaction from being shipped to a
remote region. When a task is attached for
the requested transaction, CICS checks
that the transaction is enabled in the
remote region.

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

Taskdatakey
displays the storage key in which CICS obtains all
storage for use by the transaction. This includes the task
life-time storage—the transaction work area (TWA) and
the EXEC interface block (EIB)—and the storage that
CICS obtains on behalf of programs that run under the
transaction.

The values are:

CDatakey
CICS obtains storage for the transaction from
CICS-key storage. Application programs that
execute in CICS key have read-write access to
this storage, but user-key programs have
read-only access.

Udatakey
CICS obtains storage for the transaction from
user-key storage. Application programs that
execute in any key have read-write access to this
storage.

See the description of the TASKDATAKEY parameter on
the transaction resource definition in the *CICS Resource
Definition Guide*.

Taskdataloc
displays whether certain CICS control blocks (including
EIB and TWA) for a transaction are acquired above or
below the 16MB line. The values are:

ANy The transaction accepts task-related data
anywhere.

BElow
The transaction requires any task-related data
(TWA and EIB plus any internal control blocks) to
be located below the 16MB line.

TClass**(***value***)**
displays an 8-character string identifying the name of the
transaction class to which the transaction belongs. If the
transaction does not belong to a class, DFHTCL00 is
returned.

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

To remove a transaction from its TCLASS, set this field
to DFHTCL00. An added or changed TCLASS **must** be
one that has already been defined.

TRAnsaction**(***value***)**
indicates that this panel relates to a TRANSACTION
inquiry and displays a 4-character transaction identifier.
Only transactions that have been defined in the CICS
system definition (CSD) file and installed on the running
CICS system are accessible through CEMT.

TRprof**(***value***)**
displays the name of the transaction routing profile that
defines additional options associated with this
transaction if it is defined as a remote transaction.

CEMT INQUIRE TRDUMPCODE

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.

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

- Type `CEMT INQUIRE TRDUMPCODE` (the minimum abbreviation is `CEMT I TRD`). You get a display that lists the current status.

- Type `CEMT INQUIRE TRDUMPCODE (CEMT I TRD)` followed by as many of the other attributes as are necessary to limit the range of information that you require. So, for example, if you enter `cemt i trd sy sh`, the resulting display will show you the details of only those transaction dump codes for which system dumps are to be taken and for which the CICS system is to shut down.

To change various attributes, you can:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see "Overtyping a display" on page 49).

- Use the CEMT SET TRDUMPCODE command.
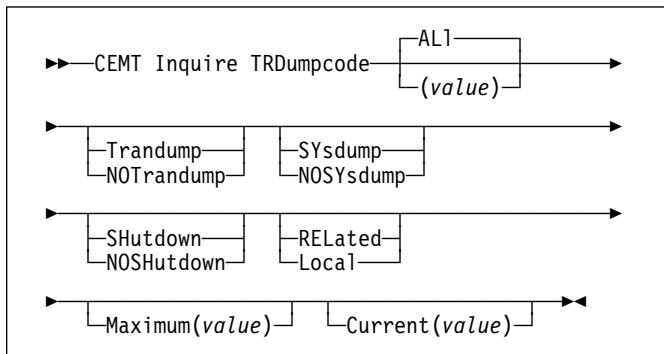
(*value***)**
is a 4-byte transaction dump code. See the *CICS Messages and Codes* manual for information on transaction dump codes.

ALl
is the default.

Sample screen

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

Figure 119. 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 120.

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

Figure 120. 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 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.

> **REL**ated
>> A system dump request is to be sent for this system dump code.

> **L**ocal
>> 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:

> **SH**utdown
>> The CICS system is to shut down.

> **NOSH**utdown
>> 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:

> **SY**sdump
>> A system dump is to be taken.

> **NOSY**sdump
>> 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:

> **T**randump
>> A transaction dump is to be taken.

> **NOT**randump
>> 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, overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see "Overtyping a display" on page 49).

```
 ►►─CEMT Inquire TSModel──┬──AL1──┬──────────────────►
                          └─(value)─┘

 ►─┬─────────────────────────┬─►◄
   ├─LOCATION(value)────────┤
   ├─POOLNAME(data-area)────┤
   ├─PREFIX(data-area)──────┤
   ├─RECOVSTATUS(value)─────┤
   ├─REMOTEPREFIX(data-area)┤
   ├─REMOTESYSTEM(data-area)┤
   └─SECURITYST(value)──────┘
```

ALl
     is the default.
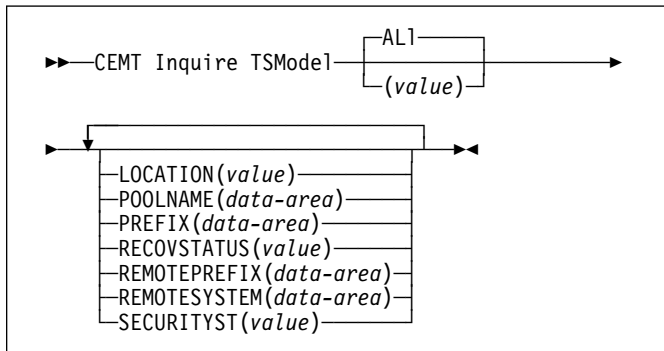
(value**)**
     is the name of the temporary storage model for which information is requested.

Sample screen

```
 I TSM
 STATUS:  RESULTS
  Tsm(DFHWEB  ) Pre(DFHWEB          ) Mai
```
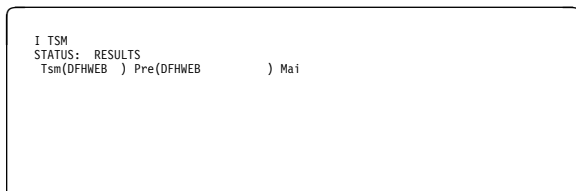
Figure 121. CEMT INQUIRE TSMODEL screen

If you place the cursor against a specific entry in the list and press ENTER, CICS displays an expanded format as shown in Figure 122.

```
 I TSM
 RESULT
  Tsmodel(DFHWEB)
  Prefix(DFHWEB)
  Location(Main)
  Recovstatus(Notrecovable)
  Securityst(Nosecurity)
  Poolname()
  Remotesystem()
  Remoteprefix()
```
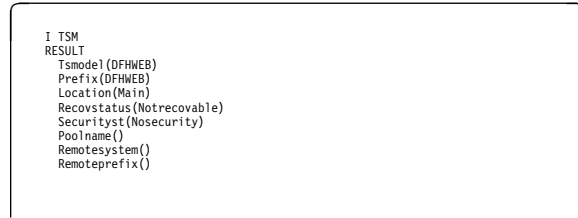
Figure 122. The expanded display of an individual entry

Displayed fields

Securityst**(**value**)**
     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.

LOcation**(**value**)**
     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**(**value**)**
     displays the recovery status of the temporary storage queue. The values are:

  Recoverable
       The temporary storage queue is recoverable.

  Notrecovable
       The temporary storage queue is not recoverable.

REMOTEPrefix**(**value**)**
     displays the remoteprefix for this model as a 16-character string.

REMOTESystem**(**value**)**
     displays the remotesystem name for this model as a 4-character string.

| **TS**model(*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 cemt i tsp
| connected, the resulting display will show you the details
| of only those shared temporary storage pools that are
| connected.

| To change various attributes, overtype your changes on the
| INQUIRE screen after tabbing to the appropriate field (see
| "Overtyping a display" on page 49).

```
                              ┌─AL1─┐
►►──CEMT Inquire TSPool───┤      ├──────────►
                              └─(value)─┘

►───────────────────────────────────►◄
        └─CONNSTATUS(cvda)─┘
```

| **AL**l
| is the default.
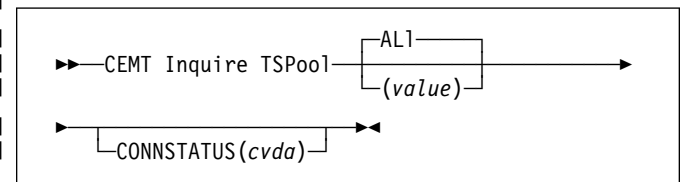
| **(**value**)**
| is the name of the temporary storage pool for which
| information is requested.

| **Sample screen**

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

| *Figure 123. 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 124.

```
 I TSP
 RESULT
  Tspool(DFHWEB)
  Connstatus(Unconnected)
```
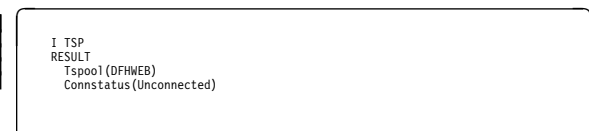
| *Figure 124. The expanded display of an individual entry*

| **Displayed fields**

| **C**onnstatus**(***value***)**
| displays the connection status of this TS pool. The
| values are:

| **C**onnected
| This TS pool is connected.

| **U**nconnected
| This TS pool is not connected.

| **T**spool**(***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:

**	BMS paging
$$	BMS route
X'fa' to X'ff'	CICS
CEBR	Default CEBR queue name
DF	CICS
DFHM	Message cache for message-protected tasks
DFxxxx	CICS REQIDS (where x is hexadecimal)

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

| • Type `CEMT INQUIRE TSQUEUE` (the minimum abbreviation
| is `CEMT I TSQ`). You get a display that lists the current
status.

• Type `CEMT INQUIRE TSQUEUE` (`CEMT I TSQ`) followed by as
many of the other attributes as are necessary to limit the
range of information that you require. So, for example, if
you enter `cemt i tsq main`, the resulting display will
show you the details of only those temporary storage
queues that are resident in main storage.

To change various attributes, overtype your changes on the
INQUIRE screen after tabbing to the appropriate field (see
"Overtyping a display" on page 49).

```
  |      ►► CEMT Inquire TSQueue ─┬─AL1─┬──────────►
  |                               └(value)┘

  ►─────────────────────────────────────────────────►
  |       ┌─SYSID(value)──┐   ┌─Numitems(value)─┐
  |       └─POOLNAME(value)┘

  ►─────────────────────────────────────────────────►
  |                └─LAstusedint(value)─┘

  ►─────────────────────────────────────────────────►
  |        └─LEngth(value)─┘  ┬─MAIn──────┬
  |                           └─AUxiliary─┘

  ►─────────────────────────────────────────────────►
  |                └─MAXitemlen(value)─┘

  ►─────────────────────────────────────────────────►
  |                └─MInitemlen(value)─┘
  |      ►──RECOVSTATUS(value)─────────────►◄
  |                      └─Transid(value)─┘
```

ALl
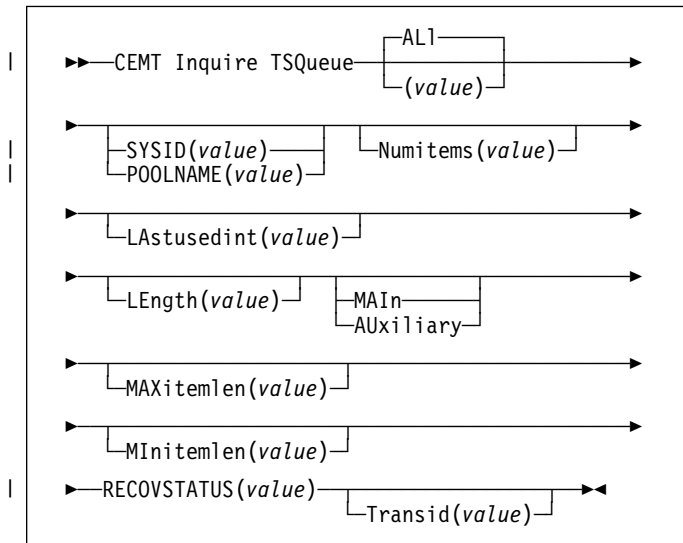
  is the default.

(value**)**

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

Sysid**(**value**)**

  specifies the system name that corresponds to the
  temporary storage pool name if the request is to be sent
  to a TS server.

  This is an input-only value, and must be specific—you
  cannot specify generic names for SYSID. If the SYSID is
  not found in a TST TYPE=SHARED entry, CICS returns
  a SYSID NOT FOUND error.

| **P**oolname**(**value**)**

| specifies the 8-character POOLNAME of a temporary
| storage pool if the request is to be sent to a TS server.

| If the POOLNAME is not found, CICS returns a
| POOLERR.

Sample screen

```
  INQ TSQ
  STATUS:  RESULTS - OVERTYPE TO MODIFY
   Tsq(AXBYQUEUENAME1 ) Num(00003) Len(0000000192) Mai
      Tra(CECI)     Max(00064) Min(00064) Las( 00001267 )
   Tsq(RDOQUEUENAME4  ) Num(00001) Len(0000000064) Aux
      Tra(CECI) Rec Max(00064) Min(00064) Las( 00000097 )
   Tsq(RDOQUEUENAME5  ) Num(00002) Len(0000000128) Aux
      Tra(CECI) Rec Max(00064) Min(00064) Las( 00000086 )
   Tsq(RDOQUEUENAME5  ) Num(00003) Len(0000000192) Mai
      Tra(CECI)     Max(00064) Min(00064) Las( 00000076 )
```

Figure 125. 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 126.

```
  INQ TSQ
  RESULT - OVERTYPE TO MODIFY
    Tsqueue(AXBYQUEUENAME1)
    Numitems(00003)
    Length(0000000192)
    Location(Main)
    Action(          )
    Transid(CECI)
    Recovstatus(Notrecovable)
    Maxitemlen(00064)
    Minitemlen(00064)
    Lastusedint(00001357)
```

Figure 126. 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 the CEBR
| transaction to browse the contents of this TS queue, (see
| Chapter 6, "CEBR—temporary storage browse" on page 21
| for details).

| Not all TS queues visible on the CEMT INQUIRE TSQUEUE
| results panel are necessarily browsable via CEBR.
| Specifically, this applies to queues resident in a shared TS
| POOL when the POOLNAME is specified on the INQUIRE
| TSQUEUE request.

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

| **Note:** For this release only you can also browse queues in
| a shared pool by specifying a SYSID which maps to
| the shared pool via an entry in the SYSID table.
| However, the SYSID required for the mapping is not
| available to CEMT, so entering 'B' next to the queue
| at first gives a NOT FOUND response from CEBR. If
| an appropriate SYSID is then provided by the user,
| the reference to the shared pool can be resolved.

| If there is no suitable entry in the SYSID table and no
| TSMODEL mapping the queue name to the shared pool, the
| queue cannot be browsed and CEBR will indicate that the
| requested queue does not exist.

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**(**value**)**

  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.

| **R**ecovstatus**(**_value_**)**
| > displays the recovery status of the temporary storage
| > queue. The values are:

| > **R**ecoverable
| > > The temporary storage queue is recoverable.

| > **N**otrecovable
| > > 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.

```
 ►►─CEMT Inquire UOW──────AL1──────────────────────►
                         └─(value)─┘

 ►───────────────────────────────────────────────►
       ┌─Backout──┐  ┌─ACtive───┐
       ├─COMmit───┤  ├─WAITIng──┤
       ├─INDoubt──┤  └─SHunted──┘
       ├─INFlight─┤
       ├─HBackout─┤
       └─HCommit──┘

 ►─────────────────────────────────────────────────►
       └─TRansid(value)─┘  └─TAskid(value)─┘

 ►─────────────────────────────────────────────────►
       └─AGe(value)─┘  └─TErmid(value)─┘

 ►─────────────────────────────────────────────────►
       └─NETName(value)─┘  └─Userid(value)─┘

 ►─────────────────────────────────────────────────►
       ┌─CONnection─┐  └─Link(value)─┘
       ├─Dataset────┤
       ├─Rlsserver──┤
       ├─WAITCommit─┤
       ├─WAITForget─┤
       └─WAITRmi────┘

 ►──────────────────────────────────────────────►◄
       └─SYsid(value)─┘  └─NETUowid(value)─┘
```

(*value*)
>   is a 16-character local identifier of a UOW.

ALl
>   is the default. If you do not specify a UOW identifier, you receive information about all UOWs in the system.

Sample screen

```
IN UOW
STATUS: RESULTS - OVERTYPE TO MODIFY
Uow(AB8FFF37584B6601) Inf Act Tra(CSSY) Tas(0000005)
   Age(00004090)                        Use(BELL   )
Uow(AB8FFF3758786C01) Inf Act Tra(CSSY) Tas(0000006)
   Age(00004090)                        Use(BELL   )
Uow(AB8FFF376D9B6601) Inf Act Tra(CSTP) Tas(0000008)
   Age(00004090)                        Use(BELL   )
Uow(AB900BD865417C04) Inf Act Tra(CSNE) Tas(0000018)
   Age(00000700)                        Use(BELL   )
Uow(AB900BE2010AC401) Inf Act Tra(CEMT) Tas(0000043)
   Age(00000689) Ter(S21D) Netn(IGBS21D) Use(BELL   )
```

Figure 127. 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 128.

```
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........)
```

Figure 128. The expanded display of an individual entry

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 netname of the remote system that caused the UOW to wait or be shunted, or blanks if the UOW is not waiting for a connection.

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.

SYsid**(*value*)**
>   displays the 4-character sysid of the connection that caused the UOW to wait or be shunted, or blanks if the UOW is not waiting for a connection or the connection has been discarded.

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 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.

TRansid**(***value***)**
    displays the 4-character ID of the transaction that started this UOW.

Uowstate
    displays the state of the UOW. The values are:

    Backout
        This UOW is being backed out, or has failed to back out one or more of the recoverable resources involved in the UOW.

    COMmit
        A decision to commit the UOW has been made, but the UOW is waiting or has been shunted. This may be because the decision has not yet been communicated to all participants in the syncpoint, or because a failure has occurred during commit processing.

    INDoubt
        This UOW is in the in-doubt state.

    INFlight
        The UOW is running normally.

    HBackout
        The UOW has been forcibly backed out. A forced decision is taken when a UOW is unable to wait for in-doubt resolution—for example, the transaction may have been defined as WAIT(NO), or have been backed out with a CEMT SET UOW command.

    HCommit
        The UOW has been forcibly committed.

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

        COMmit
            Attempt to force syncpoint commit processing, as specified for this UOW.

        Backout
            Attempt to force syncpoint backout processing, as specified for this UOW.

        Force
            Attempt to force the UOW to backout or commit, as specified on the ACTION option of the TRANSACTION resource definition.

        All these values are valid only for UOWs that have been shunted in-doubt. For information about the in-doubt attributes of TRANSACTION resource definition, see the *CICS Resource Definition Guide*.

Userid**(***value***)**
    displays the 8-character user ID with which this transaction was started.

UOW(*value***)**
    indicates that this panel relates to a UOW inquiry and displays a 16-character UOW local identifier.

Waitcause
    displays, if the UOW is waiting or shunted, the type of resource that caused it to be so.

    Note: In the case of a wait, it is the UOW that is waiting, not the task.

    The values are:

    CONnection
        This UOW is waiting or was shunted due to the failure of a session to the coordinator of the UOW during the in-doubt period. NETNAME and SYSID contain the netname and system name of the failed link.

    Dataset
        This UOW is waiting or was shunted due to the failure of one or more data sets. Use the INQUIRE UOWDSNFAIL command to identify the data sets involved and the reasons why they have caused the UOW to fail.

    RLsserver
        This UOW is waiting or was shunted due to the failure of an RLS server.

    RRms
        This UOW is waiting or has been shunted because communication has been lost with RRS/MVS.

    WAITCOmmit
        This UOW is waiting or was shunted because a failure occurred during commit processing.

    WAITForget
        This UOW is waiting for FORGET from participants in the syncpoint. Use the INQUIRE UOWLINK command to obtain the netnames and sysids of the participants.

    WAITRmi
        This UOW is waiting for FORGET from the RMI. Use the INQUIRE UOWLINK command to obtain the entry name and qualifier of the task-related user exit.

Waitstate
    displays whether syncpoint processing of the UOW has been deferred. The values are:

    ACtive
        The UOW is running normally, or is being unshunted.

    WAITIng
        Syncpoint processing has completed on this system, but not on all systems involved in the

distributed UOW. Values of WAITFORGET or
WAITRMI, and BACKOUT or COMMIT indicate
how the UOW was resolved on this system.

SHunted

Syncpoint processing of the UOW has been
deferred. SHUNTED further indicates that the task,
terminal and program storage have been released,
and locks have been retained.

CEMT INQUIRE UOWDSNFAIL

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(applb.*)`, the resulting display will
show you the details of only those failed UOWs that are
associated with data sets with a high-level qualifier of
APPLB.

CEMT INQUIRE UOWDSNFAIL

```
▶▶──CEMT Inquire UOWDsnfail──UOw(value)──────────▶
▶──DSname(value)────────────────────────────────▶
                    ├─NOtrls─┤
                    └─RLS────┘

▶─────────────────────────────────────────────▶
    ├─CAche──────┤    ├─Backupnonbwo─┤
    ├─CONnection─┤    ├─COMmitfail───┤
    ├─DATASET────┤    ├─DATASETFull──┤
    ├─RLSServer──┤    ├─DEAdlock─────┤
    └─UNdefined──┘    ├─DELexiterror─┤
                      ├─Failedbkout──┤
                      ├─INDExrecfull─┤
                      ├─IOerror──────┤
                      ├─Openerror────┤
                      ├─RLSGone──────┤
                      ├─INDOubt──────┤
                      ├─RRCommitfail─┤
                      ├─RRIndoubt────┤
                      └─Lckstrucfull─┘

▶──────────────────────────────────────────────▶◀
    └─Sysid(value)─┘    └─NEtname(value)─┘
```
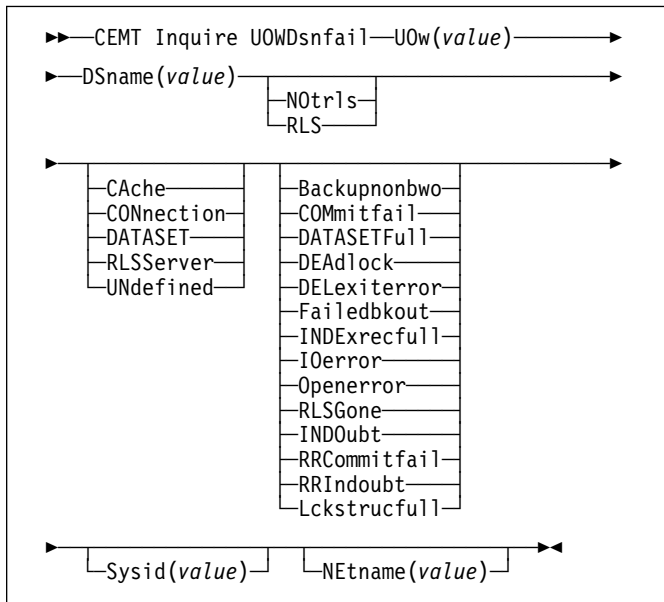
Sample screen

```
 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 129. CEMT INQUIRE UOWDSNFAIL 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 130.

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

Figure 130. The expanded display of an individual entry

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 data set, and has now failed in-doubt. The connection to the coordinating system needs to be reestablished.

IOerror
A hard I/O error occurred. To correct this error, restore a full backup copy of the data set and perform forward recovery. If you use CICSVR as your forward recovery utility, the backout is automatically retried for an RLS data set. For a non-RLS data set, use the RETRY option of the SET DSNAME command to drive the backout retry.

Lckstrucfull
An attempt to acquire a lock during backout of an update to this data set failed because the RLS lock structure was full. You must allocate a larger lock structure in an available coupling facility and rebuild the existing lock structure into it, then use the SET DSNAME (...) RETRY command to drive the backout retry.

Notapplic
The value for CAUSE is not CONNECTION, RLSSERVER, or DATASET.

Openerror
Error on opening the file for backout. A console message notifies you of the reason for the open error. One likely reason could be that the data set was quiesced.

RLSGone
An error occurred when backing out the UOW because the RLS server is down. This may also be the reason why the UOW went into backout originally. This is an error that can be resolved by recycling the server (which should happen automatically). Generally, when the server recovers, the UOWs are retried automatically. In very exceptional circumstances, it may be necessary to issue a SET DSNAME(...) RETRY command to retry UOWs that were not retried when the server returned.

RRCommitfail
An error occurred whilst RLS locks for the unit of work were being released. For this data set, the locks being released were all repeatable read locks so, if the failure was due to the RLS server being unavailable, the locks will have been released. If the failure was due to some other error from the SMSVSAM server, the locks may still be held.

RRIndoubt
The unit of work had issued repeatable read requests against the data set, and has now failed indoubt. The locks will have been released, so this failure does not prevent you from running a batch job against the data set. However, if you want to open the data set in non-RLS mode from CICS, you need to resolve the in-doubt failure before you can define the file as having RLSACCESS(NO). If the unit of work has updated any other data sets, or any other resources, you should try to resolve the in-doubt correctly. If the unit of work has only performed repeatable reads against VSAM data sets and has made no updates to other resources, it is safe to force the unit of work using the SET DSNAME or SET UOW commands.

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

Cause	Reason
CACHE	NOTAPPLIC
CONNECTION	INDOUBT
CONNECTION	RRINDOUBT
DATASET	BACKUPNONBWO
DATASET	DELEXITERROR
DATASET	DATASETFULL
DATASET	DEADLOCK
DATASET	FAILEDBKOUT
DATASET	INDEXRECFULL
DATASET	IOERROR
DATASET	LCKSTRUCFULL
DATASET	OPENERROR
RLSSERVER	COMMITFAIL
RLSSERVER	RLSGONE
RLSSERVER	RRCOMMITFAIL
UNDEFINED	Not applicable

Rlsaccess
  displays whether the data set was last opened in this CICS region in RLS or non-RLS mode. The values are:

  RLS The last open in this CICS region was in RLS mode.

  NOtrls
    The last open in this CICS region was in non-RLS mode.

Sysid(*value*)
  displays the 4-character sysid (when the cause is CONNECTION) of the remote system to which connectivity has been lost.

UOw(*value*)
  displays the 16-character local identifier of the associated unit of work.

UOWDsnfail
  indicates that this panel relates to an inquiry about the cause of a UOW failure.

CEMT INQUIRE UOWENQ

| **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 `cemt i uowe d`, the resulting display will show you the details only of enqueues on records in VSAM data sets open in non-RLS mode or CICS-maintained data tables.

```
►►──CEMT Inquire UOWEnq─────────────────────►
                       └─Uow(value)─┘

 ►──────────────────────────────────────────►
      └─TRansid(value)─┘  └─TAskid(value)─┘

 ►──────────────────────────────────────────►
     ┌─ACtive──┐  ┌─Dataset──────┐
     └─RETained─┘  ├─File─────────┤
                   ├─EXecenq──────┤
                   ├─ADdrexecenq──┤
                   ├─TDq──────────┤
                   └─TSq──────────┘

 ►──────────────────────────────────────────►
     ┌─Owner──┐  └─Netuowid(value)─┘
     └─Waiter─┘

 ►──────────────────────────────────────────►
     └─ENqfails(value)─┘  └─RESource(value)─┘

 ►──────────────────────────────────────────►
     └─RLen(value)─┘  └─QUalifier(value)─┘

 ►─────────────────►◄
     └─QLen(value)─┘
```
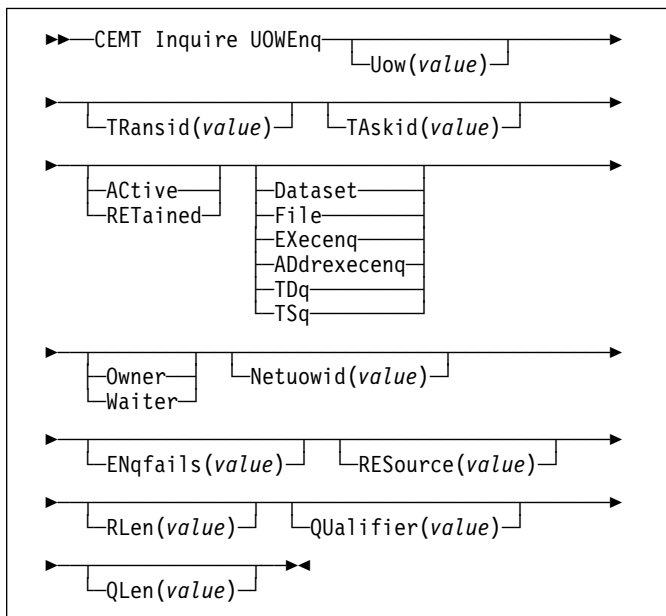
Sample screen

```
 INQ UOWE
 STATUS:  RESULTS
   Uow(AC143001D4A72601) Tra(CECI) Tas(0000038) Act Exe Own
     Res(ACCOUNTS.DBASE1      ) Rle(015) Enq(00000000)
   Uow(AC143001D4A72601) Tra(CECI) Tas(0000038) Act Add Own
     Res(g...             ) Rle(004) Enq(00000000)
```

Figure 131. CEMT INQUIRE UOWENQ screen

Note: There are blank fields on the screen where a value
does not apply or is 'negative'; that is, the value
begins with 'No'. To modify these fields, locate them
by tabbing (they appear in the same sequence as in
the expanded format), and overtype with input valid
for that field. You may find it more convenient to use
the expanded format when setting one of these
values, (negating is easier because you are
overtyping a displayed field with its negative causing
the result to become non-displayed).

If you place the cursor against a specific entry in the list and
press ENTER, CICS displays an expanded format as shown
in Figure 132.

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

Figure 132. The expanded display of an individual entry

Displayed fields

Duration**(***value***)**
displays the elapsed time (in seconds) since the
enqueue f 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 152. If no QUALIFIER data applies (that is for
EXECENQ, ADDREXECENQ, or TSQ), a value of zero
is displayed.

QLen**(***value***)**
displays the length of the data, in the range 0 through
255, displayed in the QUALIFIER field.

RELation
displays whether the data is associated with the owner
of the enqueue or with a task waiting for the enqueue.
The values are:

Owner
The (*value*), NETUOWID, TASKID, and TRANSID
are those of the owner of the enqueue.

Waiter
The (*value*), NETUOWID, TASKID, and TRANSID
are those of a waiter for the enqueue.

RLen**(***value***)**
   displays the length of the data, in the range 0 through 255, displayed in the RESOURCE field.

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.

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

State
   displays the state that the enqueue is held in. The values are:

   ACtive
      The enqueue is held in active state.

   RETained
      The enqueue is held in retained state. Its owning UOW has been shunted, or is in the process of being shunted.

TAskid**(***value***)**
   displays the number of the task associated with the UOW. If the UOW is shunted, it is the task number associated with the UOW before it was shunted.

TRansid**(***value***)**
   displays the 4-character identifier of the transaction associated with the UOW. If the UOW is shunted, it is the identifier of the transaction associated with the UOW before it was shunted.

Type
   displays the type of resource being enqueued upon. The values are:

   Dataset
      The resource is a record in a VSAM data set (or a CICS-maintained data table). RESOURCE contains the name of the data set and QUALIFIER contains the record identifier.

   File   The resource is a record in either a BDAM file or a user-maintained data table. RESOURCE contains the name of the file and QUALIFIER contains the record identifier.

      When the file is a BDAM file then the record identifier is prefixed by the BDAM block identifier. Note that truncation occurs if this combination exceeds 255 characters.

   EXecenq
      The resource is associated with an EXEC CICS ENQ request. RESOURCE contains the enqueue argument passed on the request.

   ADdrexecenq
      The resource is associated with an EXEC CICS ENQ request. RESOURCE contains the address enqueue argument passed on the request (that is,

the LENGTH parameter was omitted on the request)

TDq   The resource is a logically-recoverable transient data queue. RESOURCE contains the name of the queue. QUALIFIER contains either the string "FROMQ" or "TOQ," indicating whether a read or write lock is held for the queue.

   A READQ TD request acquires the "FROMQ" lock, whereas a WRITEQ TD request acquires the "TOQ" lock associated with the queue. A DELETEQ TD request acquires both the "TOQ" and the "FROMQ" locks.

TSq   The resource is a recoverable temporary storage queue. RESOURCE contains the name of the queue.

The data displayed in the RESOURCE and QUALIFIER fields depends on the resource type, as shown in Table 3.

Table 3. Data displayed in RESOURCE and QUALIFIER

Type	Resource	Qualifier
DATASET	Dataset name	Record identifier
EXECENQ	EXEC enqueue argument	None
ADDREXECENQ	Address of EXEC enqueue argument	None
FILE	File name	Record identifier
TDQ	TD queue name	FROMQ or TOQ
TSQ	TS queue name	None

Uow**(***value***)**
   displays the 16-character local identifier of the associated unit of work.

CEMT INQUIRE 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.
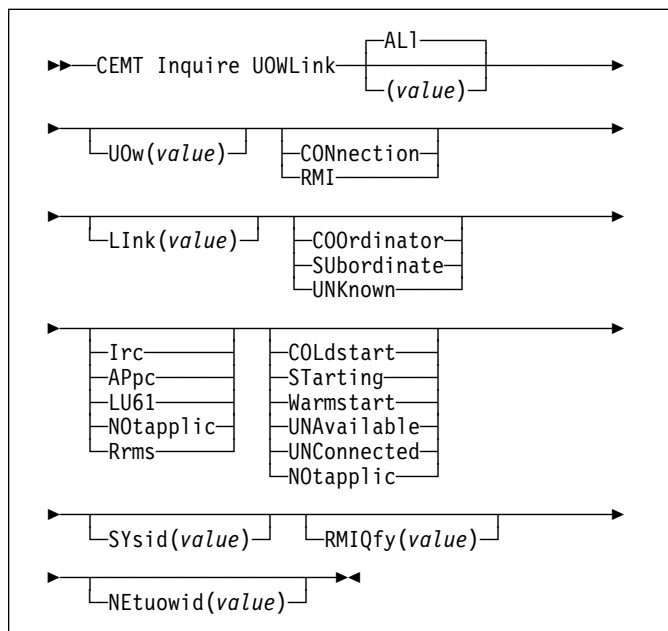
(value**)**
    is an 8-byte token identifying the UOW-link for which data is to be returned.

ALl
    is the default.

Sample screen

Figure 133. 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 134.

Figure 134. The expanded display of an individual entry

Displayed fields

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.

NEtuowid**(**value**)**
    displays a 27-character string containing the network-wide LU6.2 name for the UOW.

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

Protocol
    displays the communication protocol used by the connection. The values are:

Irc	Inter-Region Communication. This is an MRO connection.
APpc	Advanced Program to Program Communication.

| **LU**61 | LUTYPE 6.1.

| **R**rms | indicates that the UOW is coordinated by RRS/MVS.

REsyncstatus

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

COLd

The connection has been cold started by the partner system. The partner can no longer coordinate any in-doubts for this system; nor can this system pass to the partner any decisions remembered for it.

STarting

The connection is being acquired, but the exchange lognames process has not yet completed.

Ok The connection is operating normally. If there has been a failure of the partner system, the partner has been restarted and the connection able to resynchronize the associated UOW.

UNAvailable

The connection is not currently acquired.

UNConnected

There is no associated connection.

RMIQfy**(**value**)**

displays, for a connection type of RMI, the 8-character entry qualifier of the task-related user exit. For a connection type of CONNECTION, RMIQFY displays blanks.

Role

displays the role of the connection. The values are:

COOrdinator

This connection is to the syncpoint coordinator for the UOW.

SUbordinate

This connection is to a syncpoint subordinate for the UOW.

UNKnown

The syncpoint role of this connection cannot be determined.

SYsid**(**value**)**

displays the 4-character sysid of the connection. If the connection has been discarded, or the connection-type is RMI, or the PROTOCOL field is showing RRMS, SYSID displays blanks.

Type

displays the type of connection. The values are:

CFTable

A connection to a CFDT server.

CONnection

A connection defined by a CONNECTION resource definition.

RMI A connection to an external resource manager using the resource manager interface (RMI).

UOWLink**(**value**)**

indicates that this panel relates to an inquiry about a UOW-link and displays the 8-character UOW-link identifier.

UOw**(**value**)**

displays the 16-character local identifier of the associated unit of work.

| **UR**id**(**value**)**
| If the PROTOCOL field displays RRMS, this field
| contains the hexadecimal representation of the RRMS
| unit of recovery identifier. For other values of the
| PROTOCOL field, URID displays blanks.

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:

- Overtype your changes on the INQUIRE screen after tabbing to the appropriate field (see "Overtyping a display" on page 49).
- Use the CEMT SET VTAM command.

Sample screen

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

Figure 135. CEMT INQUIRE VTAM screen

If you place the cursor against the result and press ENTER, CICS displays an expanded format as shown in Figure 136.

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

Figure 136. The expanded display

```
 ►►──CEMT Inquire Vtam──►◄
```

Displayed fields

Grname(*value*)
   displays the 8-character generic resource name under which this CICS region is registered to VTAM.

Grstatus
   displays the status of generic resource registration. The values are:

   Registered
      CICS registered successfully as a member of the VTAM generic resource named by the GRNAME option.

   Deregistered
      CICS was registered as a member of the generic resource named by the GRNAME and has deregistered successfully.

   Unregistered
      A generic resource name (GRNAME) was specified, but CICS has not made any attempt to register.

   Unavailable
      A generic resource name (GRNAME) was specified, but the version of VTAM that CICS is connected to does not support generic resources.

   If CICS makes an unsuccessful attempt to register, the status is shown as UNREGISTERED, and the words REGISTER ERROR appear alongside.

   If CICS makes as unsuccessful attempt to deregister, the status is shown as REGISTERED, and the words DEREGISTER ERROR appear alongside.

Openstatus
   displays the status of the connection between CICS and VTAM. The values are:

   Open
      There is a connection between CICS and VTAM.

   Closed
      The connection between CICS and VTAM has terminated.

   Note: You can reset this value by overtyping it with a different value, or with:

      Immclose
         The connection between CICS and VTAM is in the process of closing following an EXEC CICS SET VTAM IMMCLOSE or CEMT SET VTAM IMMCLOSE command.

      Forceclose
         The connection between CICS and VTAM is in the process of closing following an EXEC CICS SET VTAM FORCECLOSE or CEMT SET VTAM FORCECLOSE command.

Psdinterval(*value*)
   displays the persistent session delay interval, expressed in the form *hhmmss*. It specifies if and for how long sessions are held in recovery-pending state after CICS fails.

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

   The permitted range is 000000 to 235959 inclusive.

   000000
      If CICS fails, sessions are terminated.

Hhmmss
>   If CICS fails, sessions are held in
>   recovery-pending state by VTAM for up to the
>   interval specified.

Vtam
>   indicates that this panel relates to a VTAM inquiry.

| **CEMT INQUIRE WEB**

| **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 W`). You will get a display
| screen, a sample of which is shown below.

| To change various attributes, you can:

| • Overtype your changes on the INQUIRE screen after
| tabbing to the appropriate field (see "Overtyping a
| display" on page 49).
| • Use the CEMT SET WEB command.

| **Sample screen**

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

| *Figure 137. CEMT INQUIRE WEB screen*

| If you place the cursor against the result and press ENTER,
| CICS displays an expanded format as shown in Figure 136
| on page 155.

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

| *Figure 138. The expanded display*
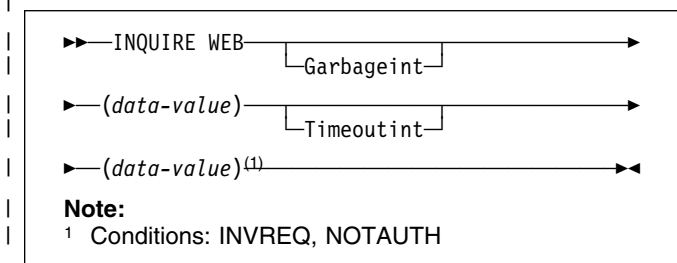
```
►►──INQUIRE WEB──────────────────────────────►
                    └─Garbageint─┘

►──(data-value)──────────────────────────────►
                    └─Timeoutint─┘

►──(data-value)⁽¹⁾────────────────────────◄
```

| **Note:**
| [1] Conditions: INVREQ, NOTAUTH

| **Displayed fields**

| **GARBAGEINT**
| is the interval at which the Web garbage collection task
| runs to clean up Web 3270 state data for which the
| terminal timeout interval has expired. The interval is
| measured in minutes.

| **TIMEOUTINT**
| is the period of time after which inactive Web 3270
| sessions are eligible for garbage collection. The period is
| measured in minutes.

CEMT PERFORM commands

The CEMT PERFORM commands are not concerned with
resource status. They are used to:

- Cause the CICS timeout delete mechanism to be
  invoked immediately (PERFORM DELETSHIPPED)

- Dump and continue processing (PERFORM DUMP or
  PERFORM SNAP)

| - End an affinity owned by CICS (PERFORM
| ENDAFFINITY)

- 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 DELETSHIPPED

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

PERFORM DELETSHIPPED invokes timeout delete immediately.

Notes:

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

2. Issuing a PERFORM DELETSHIPPED command does **not** reset the start of the time interval until the next invocation of the timeout delete mechanism. In other words, causing the timeout delete mechanism to be invoked immediately does not affect the time remaining until the next scheduled invocation.

Syntax

```
►►──CEMT Perform DEletshipped──►◄
```
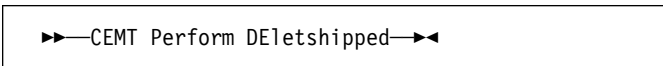
CEMT PERFORM DUMP|SNAP

Function: Obtain a CICS system dump.

Description: To get a CICS system to dump and continue processing, you can use either the CEMT PERFORM DUMP command or the CEMT PERFORM SNAP command. Both commands produce a "snapshot" of the CICS system.

The use of CEMT PERFORM DUMP or CEMT PERFORM SNAP prevents all other tasks from running. Consequently, terminal operators may experience delays in response.

Syntax

```
►►──CEMT Perform──┬─DUmp─┬──────────────────►◄
                  └─SNap─┘  └─Title(value)─┘
```
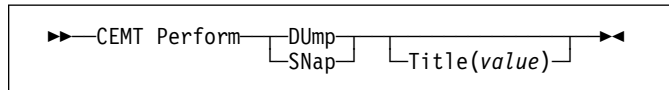
Options

DUmp
    specifies that a system dump is to be produced.

SNap
    specifies that CICS is to issue the MVS SDUMP macro to produce an SDUMP (system dump) of the CICS region.

Title**(**value**)**
    specifies a title of up to 32 characters; it is added at the beginning of the dump. If your title includes spaces, you must enclose the whole title within single quotation marks (' ').

CEMT PERFORM ENDAFFINITY

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──NETName(value)──►

►──────────────────────►◄
    └─NETId(value)─┘
```
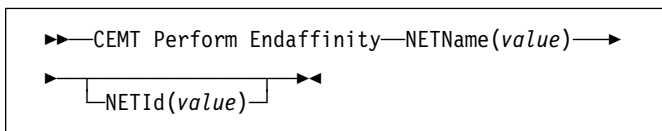
Options

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 RESET

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

This command should be executed following a change to the system date or to the time-of-day.

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.

CICS generally obtains and stores the local time at specific times of day only (for example, at start up, and midnight). Thus you should execute this command whenever you change the system date or time-of-day while CICS is running. Note that to obtain the time, CICS issues an MVS STCK macro and modifies this by the local time difference, if any. Whenever an application program issues an EXEC CICS ASKTIME command, CICS updates the EIBTIME field in the exec interface block, always with the local time. To ensure that your applications are receiving the correct time, use CEMT PERFORM RESET whenever you alter the MVS time.

Syntax

```
►►──CEMT Perform Reset──►◄
```
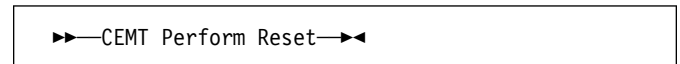
CEMT PERFORM SECURITY

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

```
►►──CEMT Perform SEcurity──────────────►◄
                        └─Rebuild─┘
```
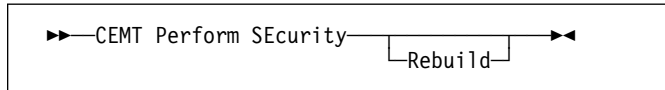
If the external security manager is RACF 2.1 or an equivalent product, the PERFORM SECURITY command is not required to refresh the classes used for resource checks. A message of NOT REQUIRED is displayed and no action is taken.

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.

Syntax

```
►►──CEMT Perform SHUTdown──────────────────────►
                        └─Xlt(value)─┘

►──────────────────────────────────────────────►
   └─Plt(value)─┘  ┌─Sdtran(value)─┐
                   └─NOSdtran──────┘

►──────────────────────────────────────────────►
   └─Immediate──────────────┘
               └─NORestart─┘

►──────────────────────────────────────►◄
   └─Takeover─┘   └─Dump─┘
```
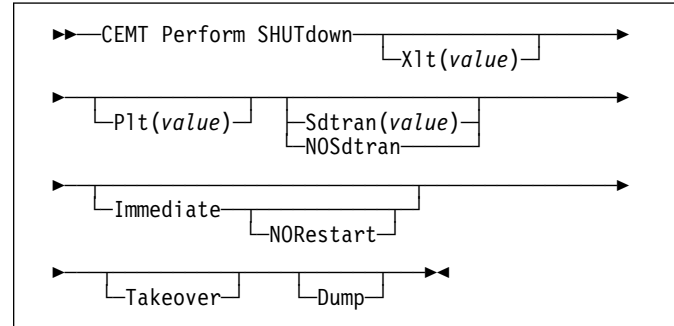
Options

Dump
   A dynamic storage dump is produced upon completion of the termination process.

Immediate
   specifies that the system is shut down immediately, terminating all active tasks and SNA sessions. If IMMEDIATE is not specified, all tasks are allowed to finish, and SNA sessions are allowed to terminate normally.

NORestart
   specifies that if the CICS region is registered to
\#   MVS/ESA Automatic Restart Manager (ARM),
\#   it is to be deregistered as part of shutdown and not
\#   restarted automatically.

NOSdtran
   No shutdown assist transaction is to be run at CICS shutdown.

   The default shutdown transaction, if neither SDTRAN nor NOSDTRAN are specified, is CESD.

Plt(*value*)
   specifies the 2-character suffix of a program list table (PLT) to be used for the shutdown.

Sdtran
   The 4-character identifier of the shutdown assist transaction

Takeover
   specifies that this CICS region should be closed down, and the alternate CICS region is to take over. This is valid only if the XRF=YES system initialization parameter was specified for CICS startup.

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

Transactions listed in the specified XLT can be initiated after the SHUTDOWN request, and before the system quiesces. The default XLT should contain an entry for the master terminal transaction. This ensures that you can issue an immediate shutdown if, for example, a task is found to be suspended after an ordinary (nonimmediate) shutdown has been requested.

Table 4 shows the effect of the TAKEOVER and IMMEDIATE options on the active CICS system.

Table 4. Effect of the TAKEOVER and IMMEDIATE options on the active CICS system			
	CEMT PERFORM SHUTDOWN		
	(normal)	*TAKEOVER*	*IMMEDIATE*
Tasks	finish	finish	terminate
SNA session (XRF)	unbind	leave	leave
SNA session (non-XRF)	unbind	unbind	unbind

Table 5 shows the effect of the TAKEOVER and IMMEDIATE options on the alternate CICS system.

Table 5. Effect of the TAKEOVER and IMMEDIATE options on the alternate CICS system			
	CEMT PERFORM SHUTDOWN		
	(normal)	*TAKEOVER*	*IMMEDIATE*
CICS alternate	shuts down	takes over	takes over

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

CEMT PERFORM STATISTICS

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 113 and page "CEMT SET STATISTICS" on page 197 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.

Syntax

```
►►──CEMT Perform STatistics──────────────────►
                              └─RECORd─┘

  ┌──────────────────────────┐
►─┴─ALl──────────────────────┴─►◄
     └─RESetnow─┘
  ──AUtoinstall──
  ──Connection──
  ──DB2──
  ──DIspatcher──
  ──Enqueue──
  ──FEpi──
  ──FIle──
  ──Journalname──
  ──Lsrpool──
  ──Monitor──
  ──PROGAuto──
  ──PROGRam──
  ──RECOVery──
  ──STAts──
  ──STOrage──
  ──STReamname──
  ──SYsdump──
  ──TAblemgr──
  ──TCLass──
  ──TCPipservice──
  ──TDqueue──
  ──TErminal──
  ──TRAnsaction──
  ──TRDump──
  ──TSqueue──
  └─Vtam──
```
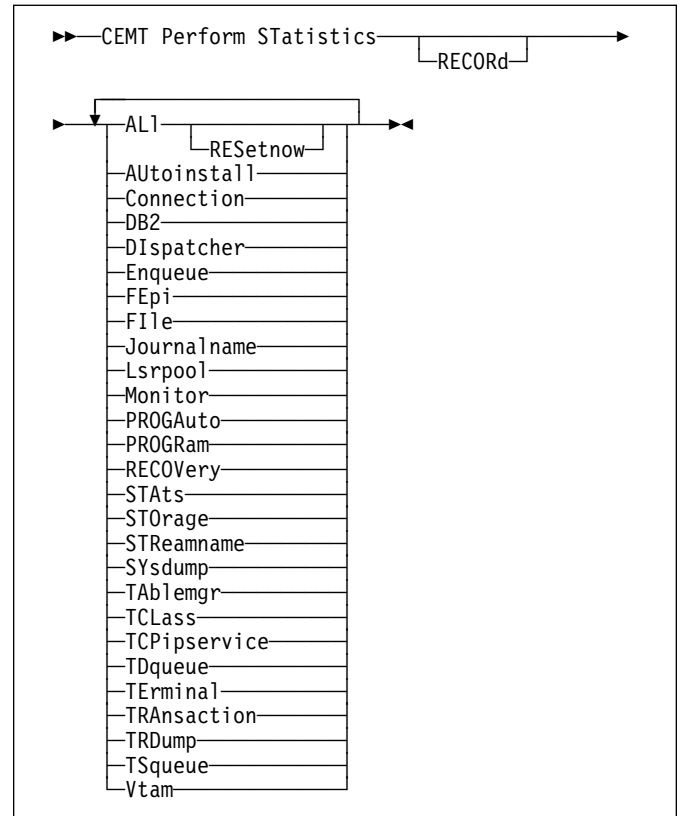
CEMT PERFORM STATISTICS

Options

ALl
Statistics for all resources are to be written immediately to the SMF data set.

ALL RESETNOW resets the counters for the collection of statistics. RESETNOW only works if the ALL operand is set, that is, ALL RESETNOW.

AUtoinstall
Autoinstall statistics are to be written immediately to the SMF data set.

Connection
Connection 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.

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.

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.

RESetnow
The statistics counters are to be reset.

STAts
Statistics domain statistics are to be written immediately to the SMF data set.

STOrage
Storage manager statistics are to be written immediately to the SMF data set.

STReamname
Log stream statistics are to be written immediately to the SMF data set.

SYsdump
System dump statistics are to be written immediately to the SMF data set.

TAblemgr
Table manager statistics are to be written immediately to the SMF data set.

| **TCL**ass
Transaction class statistics are to be written immediately to the SMF data set.

| **TCP**ipservice
| TCP/IP service statistics are to be written immediately to
| the SMF data set.

TDqueue
Transient data statistics are to be written immediately to the SMF data set.

TErminal
Terminal statistics are to be written immediately to the SMF data set.

TRAnsaction
Transaction statistics are to be written immediately to the SMF data set.

TRDump
Transaction dump statistics are to be written immediately to the SMF data set.

TSqueue
Temporary storage statistics are to be written immediately to the SMF data set.

Vtam
VTAM 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

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 and MVS consoles. For information about autoinstall, see the *CICS Resource Definition Guide*.

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

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

- Type CEMT SET AUTOINSTALL (CEMT S AUTO) followed by one or more attribute settings that you wish to change. For example, cemt s aut m(newvalue) p(newname) resets the values for maximum number of requests for a new program.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

```
►►──CEMT Set AUToinstall──┬──────────┬──────►
                          ├─PROGAUTO─┤
                          ├─FULLAUTO─┤
                          └─NOAUTO───┘
   ►──┬────────────────┬──┬──────────────────┬──►◄
      └─Maxreqs(value)─┘  └─Program(value)──┘
```
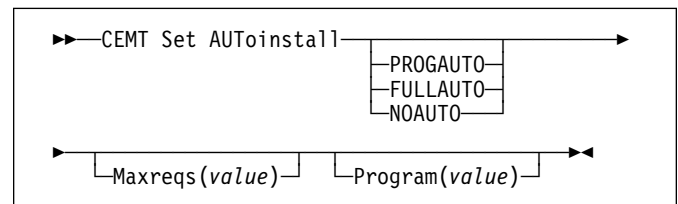
Options

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

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 16, "CETR—trace control" on page 225.

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

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

- Type CEMT SET AUXTRACE (CEMT S AUX) followed by one or more attribute settings that you wish to change. For example, cemt s aux sto a stops auxiliary tracing and causes automatic switching to occur as necessary.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

```
►►──CEMT Set AUXtrace──────────────────────────►
                      ├─STArt─┤
                      ├─Pause─┤
                      └─STOp──┘

 ►──────────────────────────────►◄
      ├─NOswitch─┤  └─SWitch─┘
      ├─NExt─────┤
      └─All──────┘
```

Options

All

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

NExt

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

NOswitch

Switching between the auxiliary trace data sets is not to occur without operator intervention.

Pause

Auxiliary tracing is to stop, but the data set is to remain open. A subsequent START request causes trace entries to be written immediately following those that were written before the PAUSE request.

STArt

CICS is to start auxiliary tracing and open the auxiliary trace data set if it is currently closed.

STOp

CICS is to stop auxiliary tracing and close the auxiliary trace data set. A subsequent START request causes new trace entries to be written at the start of the data set, thereby overwriting the trace entries that were written before the STOP request.

SWitch

specifies that an immediate switch from the current auxiliary trace data set is to be made to the alternate trace data set.

CEMT SET 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 overtype them with the required values.

* Type `CEMT SET CONNECTION` (`CEMT S C`) followed by one or more connection identifiers or ALL, followed in turn by one or more attribute settings that you wish to change. For example, `cemt s c al i` resets the values for all connections to make them available for use (inservice).

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

```
►►──CEMT Set Connection──┬─(value)─┬──────────►
                         └─ALl─────┘

►──┬───────────────┬──┬─INService──┬──────────►
   └─NOTpending────┘  └─OUtservice─┘

►──┬───────────┬──┬─PUrge───────┬─────────────►
   ├─ACquired──┤  ├─FORCEPurge──┤
   └─RELeased──┘  ├─CAncel──────┤
                  └─FCancel─────┘

►──┬───────────┬──┬─ENdaffinity─┬─────────────►
   ├─COmmit────┤  └─────────────┘
   ├─Backout───┤
   ├─FORCEUow──┤
   └─RESync────┘

►──┬──────────────┬──►◄
   └─NORecovdata──┘
```
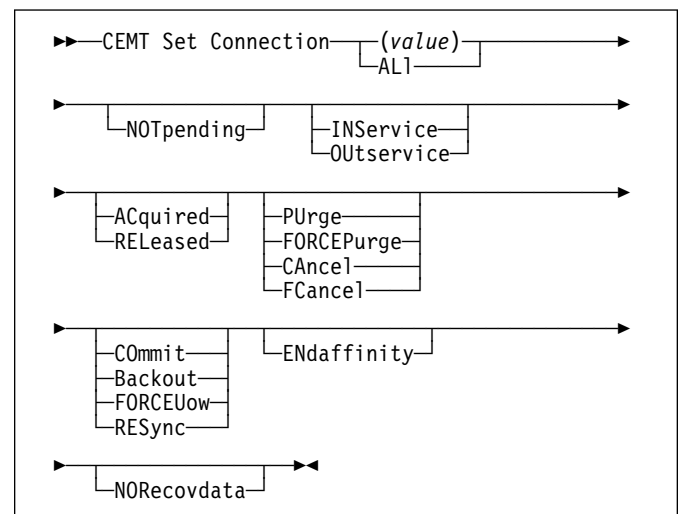
Options

(*value***)**

One or more names (1–4 characters) defined for an interregion communication (IRC) or an intersystem communication (ISC) connection.

CEMT SET CONNECTION

ACquired **(***appc only***)**

specifies whether CICS is to acquire a session with the logical unit represented by the CONNECTION name. To get more detailed information about the availability status of the connection elements, use the CEMT INQUIRE MODENAME command. A connection cannot be both ACQUIRED and OUTSERVICE.

ALl

Any changes you request are made to all resources of the specified type that you are authorized to access.

Backout **(APPC parallel-session, CICS-to-CICS MRO, and LU61 only)**

All UOWs shunted due to the failure of this connection are to be backed out. The normal resynchronization process is to be partially overridden: decisions are taken for any units of work that are in-doubt due to a failure of the connection; but the decisions are recorded and any data inconsistencies are reported when the connection is next acquired.

CAncel

Automatic initiate descriptors (AIDs) queuing for the specified connection are to be canceled.

AIDs representing scheduled and allocated requests waiting in the local CICS system for the specified connection are canceled. However, TD AIDs with an associated triggered task already started are not be canceled. In addition, the following CICS system AIDs are not purged unless FCANCEL is specified.

Table 6. 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.

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 6 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.

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 OS/390 partner that has performed an initial start

- A connection to a pre-CICS Transaction Server for OS/390 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 OS/390 partner, a lognames mismatch indicates that the partner has performed an initial start. For a pre-CICS Transaction Server for OS/390 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 168.

2. NOTPENDING has no effect on MRO connections to pre-CICS Transaction Server for OS/390 systems. However, the cold start of a pre-CICS Transaction Server for OS/390 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 OS/390 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 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.

```
►►──CEMT Set DB2Conn──┬──────────────────────────────────┬──┬────────────────────────────────────┬──►◄
                      │        ┌────────────────────┐     │  │        ┌────────────────────────┐    │
                      │        ├─UOW──────┐          │     │  │        ├─DB2ID(data-area)───────┤    │
                      │        ├─TASK─────┤          │     │  │        ├─MSGQUEUE1(data-area)────┤    │
                      │        ├─TXID─────┤          │     │  │        ├─MSGQUEUE2(data-area)────┤    │
                      │        └─NONE─────┘          │     │  │        ├─MSGQUEUE3(data-area)────┤    │
                      │        ┌─AUTHID(data-area)───┐     │  │        ├─RELEASE────┐            │    │
                      │        │  ├─GROUP─────┐       │     │  │        └─NORELEASE──┘            │    │
                      │        │  ├─SIGN──────┤       │     │  │        ┌─PLAN(data-area)─────────┐ │    │
                      │        │  ├─TERM──────┤       │     │  │        └─PLANEXITNAME(data-area)─┘ │    │
                      │        │  ├─TX────────┤       │     │  │        ├─HIGH──────┐             │    │
                      │        │  ├─OPID──────┤       │     │  │        ├─EQUAL─────┤             │    │
                      │        │  └─USERID────┘       │     │  │        └─LOW───────┘             │    │
                      │        ┌─COMAUTHID(data-area)─┐     │  │        ├─PURGECYCLEM(data-area)──┤    │
                      │        │  ├─CGROUP────┐       │     │  │        ├─PURGECYCLES(data-area)──┤    │
                      │        │  ├─CSIGN─────┤       │     │  │        ├─SECURITY(REBUILD)───────┤    │
                      │        │  ├─CTERM─────┤       │     │  │        ├─SIGNID(data-area)───────┤    │
                      │        │  ├─CTX───────┤       │     │  │        ├─NOCONNECT──┐            │    │
                      │        │  ├─COPID─────┤       │     │  │        ├─CONNECT────┤            │    │
                      │        │  └─CUSERID───┘       │     │  │        └─RECONNECT──┘            │    │
                      │        ├─COMTHREADLIM(data-area)    │  │        ├─STATSQUEUE(data-area)───┤    │
                      │        ├─ABEND─────┐          │     │  │        ├─TCBLIMIT(data-area)─────┤    │
                      │        └─SQLCODE───┘          │     │  │        ├─THREADLIMIT(data-area)──┤    │
                      │        ├─CONNECTED──────────┐ │     │  │        ├─TWAIT──────┐            │    │
                      │        ├─NOTCONNECTED───────┤ │     │  │        ├─NOTWAIT────┤            │    │
                      │        └─FORCENOTCONNECTED──┘ │     │  │        └─N906───────┘            │    │
                      └──────────────────────────────┘     │  └────────────────────────────────────┘
```

Options

Accountrec
specifies the minimum amount of DB2 accounting required for transactions requiring pool threads. The specified minimum can be exceeded, as described in the following options. CVDA values are:

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 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 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 can 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, DB2 uses the connected group name supplied by the CICS attachment facility as the secondary DB2 authorization ID. If the RACF list of group options is active, DB2 ignores the connected group name supplied by the CICS attachment facility, but the value appears in the DB2 list of secondary DB2 authorization IDs.

To use the GROUP option the CICS system must have SEC=YES specified in the CICS system initialization table (SIT).

If no RACF group ID is available for this USERID, an 8-character field of blanks is passed to DB2 as the group ID.

SIGN
Specifies the SIGNID parameter of the DB2CONN should be used as the resource authorization ID.

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

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

TX   Specifies the transaction identification (four characters padded to eight) as the authorization ID.

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

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

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

COMAUTHId*(data-area)*
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, 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 DSNC STRT to start the CICS DB2 attachment. If the requested DB2 subsystem is active, control returns when CICS and DB2 have been connected. If the requested DB2 subsystem is not active, the response returned is dependant on the setting of STANDBYMODE: If DB2 is not initialized, and STANDBYMODE(NOCONNECT) is specified on the DB2CONN, INVREQ and RESP2=39 is returned. If you specify STANDBYMODE(CONNECT) or STANDBYMODE(RECONNECT), NORMAL with RESP2=38 is returned indicating that the CICS DB2 attachment is in standby mode and will connect to DB2 as soon as it becomes active.

NOTCONNECTED
> CEMT initiates a quiesce stop of the CICS DB2 connection and implicitly uses the NOWAIT option, meaning that control returns immediately to CEMT before the quiesce is complete.

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

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.

MSGQUEUE1
> specifies the first transient data destination to which unsolicited messages from the CICS DB2 attachment are sent.

MSGQUEUE2
> specifies the second transient data destination to which unsolicited messages from the CICS DB2 attachment are sent.

MSGQUEUE3
> specifies the third transient data destination to which unsolicited messages from the CICS DB2 attachment are sent.

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

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

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

PLAN
> specifies the name of the plan to be used for all threads in the pool. If PLAN is specified, PLANEXITNAME 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 subtasks relative to the CICS main task (QR TCB). CVDA values are:

HIGH
> subtasks attain a higher priority than the CICS main task from which the subtask was generated.

EQUAL

subtasks have equal priority with the CICS main task.

LOW

subtasks have a lower priority than the CICS main task.

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.

SECURITY(REBUILD)

specifies that the CICS DB2 attachment should force all existing threads to sign-on again at the next thread reuse. It should be used when RACF profiles have been updated by issuing the following commands:

- CEMT PERFORM SECURITY REBUILD for RACF 1.9.2 or earlier
- TSO SETROP TS RACLIST(xxxxxxxx) REFRESH for RACF 2.1 or later

SIgnid

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

Standbymode

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

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 subtasks that can be identified to 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 SIgnid, a surrogate user security check is invoked if security is active. This ensures that the userid under which SET is being executed is authorized to act on behalf of the userid being set.

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.

```
►►──CEMT Set DB2Entry──────────────────────────►
                      └─(data-area)─┘

►─┬──────────────────────────────┬─►◄
  │ ┌──────────────────────────┐ │
  └─┬─UOW─────────────────────┬─┘
    ├─TASK────────────────────┤
    ├─TXID────────────────────┤
    └─NONE────────────────────┘
    ├─AUTHID(data-area)───────┤
    │ ┌─GROUP─────────────────┤
    ├─SIGN────────────────────┤
    ├─TERM────────────────────┤
    ├─TX──────────────────────┤
    ├─OPID────────────────────┤
    └─USERID──────────────────┘
    ├─ABEND───────────────────┤
    ├─SQLCODE─────────────────┤
    └─POOL────────────────────┘
    ├─ENABLED─────────────────┤
    ├─DISABLED────────────────┤
    └─FORCEDISABLED───────────┘
    ├─PLAN(data-area)─────────┤
    └─PLANEXITNAME(data-area)─┘
    ├─HIGH────────────────────┤
    ├─EQUAL───────────────────┤
    └─LOW─────────────────────┘
    ├─PROTECTNUM(data-area)───┤
    ├─THREADLIMIT(data-area)──┤
    ├─TWAIT───────────────────┤
    ├─NOTWAIT─────────────────┤
    └─TPOOL───────────────────┘
```

Options

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

UOW
The CICS DB2 attachment facility causes an accounting record to be produced for each UOW, assuming that the thread is released at the end of the UOW.

TASK
The CICS DB2 attachment facility causes a minimum of one accounting record to be produced for each CICS task.

A transaction containing multiple UOWs can use a different thread for each of its UOWs (assuming the thread is released at syncpoint). The result can be an accounting record produced for each UOW.

TXID
The CICS DB2 attachment facility produces at least one accounting record to be produced when the transid using the thread changes.

This option applies to DB2ENTRYs that are used by more than one transaction ID. As threads are typically released st syncpoint, a transaction containing multiple UOWs can use a different

thread for each UOW. The result can be that an accounting record is produced per UOW.

NONE

No accounting records are required for transactions using threads from this DB2ENTRY.

DB2 does, however, produce at least one accounting record for each thread when the thread is terminated. Additionally, authorization changes cause accounting records to be produced.

AUTHID

specifies the id to be used for security checking when using this DB2ENTRY. If AUTHId is specified, AUTHTYPE is cleared.

Authtype

returns the type of id that can be used for security checking when using this DB2ENTRY. If AUTHTYpe is specified, AUTHid is cleared. CVDA values are:

GROUP

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

IDs passed to DB2	How DB2 interprets values
CICS sign-on user ID (USERID)	Represents the primary DB2 authorization ID.
RACF-connected group name	If the RACF list of group options is not active, DB2 uses the connected group name supplied by the CICS attachment facility as the secondary DB2 authorization ID. If the RACF list of group options is active, DB2 ignores the connected group name supplied by the CICS attachment facility, but the value appears in the DB2 list of secondary DB2 authorization IDs.

To use the GROUP option the CICS system must have RACF external security SEC=YES specified in the CICS system initialization table (SIT).

If no RACF group ID is available for this USERID, an 8-character field of blanks is passed to DB2 as the group ID.

SIGN

Specifies the SIGNID parameter of the DB2CONN should be used as the resource authorization ID.

TERM

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

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

TX Specifies the transaction identification (four characters padded to eight) as the authorization ID.

OPID

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

USERID

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

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

Disabledact

specifies what CICS is to do with new transactions that access a DB2ENTRY when it has been disabled or is disabling. CVDA values are:

POOL

The CICS DB2 attachment facility routes the request to the pool. Message DFHDB2072 is sent to the transient data destination specified by MSGQUEUEn on the DB2CONN for each transaction routed to the pool.

ABEND

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

SQLCODE

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

Enabledstatus

specifies whether the DB2ENTRY can be accessed by applications. CVDA values are:

ENABLED

The DB2ENTRY can be accessed by applications.

DISABLED

The DB2ENTRY cannot be accessed by applications. A DB2ENTRY has to be disabled before it can be reinstalled or discarded.

FORCEDISABLED

All tasks using the DB2ENTRY and those queued against the DB2ENTRY are forcepurged. The DB2ENTRY is then disabled.

PLAN

specifies the name of the plan to be used for this DB2ENTRY.

If PLAN is specified, PLANEXITNAME is cleared.

PLANEXITNAME

specifies the name of the dynamic plan exit to be used for this DB2ENTRY. If you change the PLAN and PLANExitname while there are active transactions for the DB2ENTRY the next time the transaction releases the thread, the plan/exit is determined using the new rules. If PLANExitname is specified, PLAN is cleared.

Priority

specifies the priority of the thread tasks for the DB2ENTRY relative to the CICS main task (QR TCB). CVDA values are:

HIGH

subtasks attain a higher priority than the CICS main task from which the subtask was generated.

EQUAL

subtasks have equal priority with the CICS main subtask.

LOW

subtasks have a lower priority than the CICS main task.

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 overtype them with the required values.

- Type `CEMT SET DB2TRAN` (`CEMT S DB2T`) with either the name of a DB2TRAN or ALL, followed by one or more attribute settings that you wish to change. For example, `cemt s db2t all db2e(entry1)` will set all DB2TRANs to refer to a DB2ENTRY named entry1.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

```
►►──CEMT Set DB2Tran──────────────────────────►

   ┌◄─────────────────────────┐
   ▼                          │
───┼──────────────────────────┼──►◄
   ├─DB2Entry(data-area)─┤
   └─Transid(data-area)──┘
```
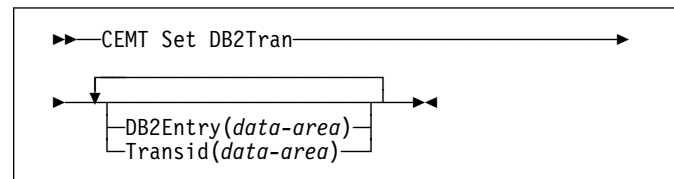
Options

DB2Entry

specifies the name of the DB2ENTRY to which this DB2TRAN refers; that is, the DB2ENTRY with which this additional transid should be associated.

Transid

specifies the transaction id to be associated with the entry. The transaction id can include wildcard characters (see the *CICS Resource Definition Guide* for information about use of wildcard characters). If you change TRANSID for a DB2TRAN while the attachment is active, all transactions with a thread continue to use the thread from that entry until it is released for reuse. When that transaction issues the next SQL statement, the thread is acquired from the entry or pool based upon the new definition.

DB2TRAN parameters can be set at any time.

CEMT SET DELETSHIPPED

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 DELETSHIPPED allows you to change the values that control the mechanism.

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

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

- Type `CEMT SET DELETSHIPPED` (`CEMT S DE`), followed by one or more attributes that you wish to change. For example, `cemt s de idl(0150000)` specifies that shipped terminal definitions are to remain installed for at least fifteen hours after they become inactive.

```
►►──CEMT Set DEletshipped────────────────────────►

►─────────────────────────────────────────────►◄
       └─INterval(value)─┘   └─IDle(value)─┘
```
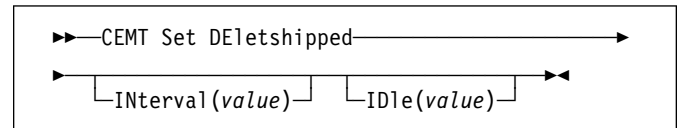
Options

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 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──────────────────────────────────►
                   └─Dsalimit(value)─┘

►──────────────────────────────────────────►◄
    └─EDsalimit(value)─┘
```
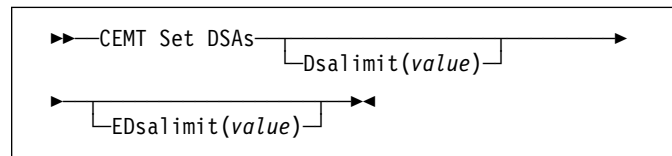
Options

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 could fail.

Some of the attributes of a data set cannot be set until the first file that references the data set has been opened. Where an attribute is not valid until a file has been opened, an error is returned. Note that QUIESCESTATE is an attribute that can be set before any files have been opened against the specified data set.

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

- Type `CEMT SET DSNAME` (the minimum abbreviation is `CEMT S DSN`) with either a value corresponding to a data set name or 'ALL'. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE DSNAME. You can then tab to the highlighted or blank fields and overtype them with the required values.

- Type `CEMT SET DSNAME` (`CEMT S DSN`) with either a dataset_name or 'ALL', followed by one or more attributes that you wish to change. For example, `cemt s dsn(applb.*) q` quiesces all data sets with a high-level qualifier of `APPLB` in all CICS regions within the sysplex.

Resetting the values takes effect immediately.

```
►►──CEMT Set DSName──┬──(value)──┬─────────►
                     └─ALl───────┘

►──┬──────────────┬──┬──────────────┬──────►
   │─REMOve───────│  │─AVailable────│
   │─RECOVEREd────│  │─UNAvailable──│
   │─RESetlocks───│
   │─RETRy────────│

►──┬──────────────┬──┬──────────┬──►◄
   │─Quiesced─────│  │─BACkout──│
   │─UNQuiesced───│  │─Commit───│
   │─IMmquiesced──│  │─FOrce────│
```
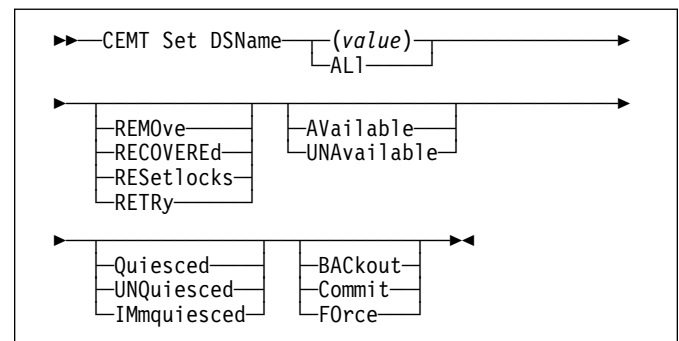
Options

(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.

CEMT SET DSNAME

IMmquiesced

Causes all existing CICS open RLS-mode files
throughout the sysplex to be closed and the data set to
be marked as quiesced in the ICF catalog. Each CICS in
the sysplex abends all in-flight UOWs that are accessing
the data set before closing files, causing in-flight UOWs
to back out. Any UOWs that fail backout are shunted. No
RLS-mode files can open against this data set, but
non-RLS-mode open requests are permitted.

In addition to closing open files, IMMQUIESCED sets the
file state to UNENABLED if it was ENABLED. A
subsequent SET DSNAME UNQUIESCED command
restores the file state to ENABLED, if it was set
UNENABLED by a QUIESCED or IMMQUIESCED
action, but *not* if the UNENABLED state is due to some
other event. This state change is recorded in the CICS
global catalog.

> ### 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 can
> 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 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.

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

Function: Set the status of enqmodel definitions on the local system.

Description: Each ENQMODEL definition can be in one of the following states

ENABLED Matching ENQ requests will be processed in the normal way.
DISABLED Matching ENQ requests will be rejected, and the issuing task abended abcode ENQ_DISABLED. Matching INSTALL/CREATE/DISCARD requests will be processed.
WAITING Matching ENQ requests will be rejected, and the issuing task abended abcode ENQ_DISABLED. There are INSTALL/CREATE/DISCARD requests waiting to be processed.

This command provides a means for you to set the ENABLED and DISABLED states for ENQmodel resources installed on the local system. An enqmodel must be enabled to allow matching EXEC ENQ requests to be processed. It must be disabled to allow a more specific enqmodel to be enabled.

Enqmodels forming nested generic enqnames must be enabled in order, from the most to the least specific. For example, enable ABCD* then ABC* then AB*. If you attempt to enable a more specific enqmodel when a less specific enqmodel is already enabled, the result is that message DFHNQ0107 is issued.

You cannot enable/disable an enqmodel which is in the waiting state.

Syntax

```
►►──CEMT Set ENQModel(value)──────────────►◄
                            ├─Enabled──┤
                            └─DIsabled─┘
```
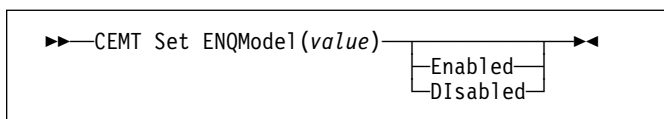
Options

(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

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 overtype them with the required values.

- Type CEMT SET FILE (CEMT S FI) with either the name of a file or ALL, followed by one or more attribute settings that you wish to change. For example, cemt s fi al op en attempts to set all files open and enabled for use.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Reset values take effect immediately.

Note: Some attributes can be changed only when the specified file is closed and disabled. In these cases, the reset values take effect when the file is enabled for use again.

```
►►──CEMT Set FIle──┬─(value)─┬──────────────►
                   └─ALl────┘

►──┬─OPen───────┬──┬─ENabled──┬──┬─REAd───┬──►
   ├─CLosed─────┤  └─DIsabled─┘  └─NORead─┘
   └─Forceclose─┘

►──┬─UPdate───┬──┬─ADdable───┬──────────────►
   └─NOUpdate─┘  └─NOAddable─┘

►──┬─BRowse───┬──┬─DElete───┬───────────────►
   └─NOBrowse─┘  └─NODelete─┘

►──┬─EXclusive───┬──┬─EMptyreq───┬──────────►
   └─NOEXclusive─┘  └─NOEMptyreq─┘

►──┬─Old───┬──┬─RLs────┬──┬─DSname(value)─┐──►
   └─Share─┘  └─NOTRls─┘

►──┬─NOTTable──┬──┬─LOAd───┐────────────────►
   ├─CICstable─┤  └─NOLoad─┘
   ├─USertable─┤
   └─CFTable───┘

►──┬─CFDtpool(value)─┐──────────────────────►

►──┬─Tablename(value)─┐──┬─COntention─┬──────►
                        └─LOCking────┘

►──┬─Maxnumrecs(value)─┐────────────────────►

►──┬─Keylength(value)─┐─────────────────────►

►──┬─RECordsize(value)─┐───►◄
```
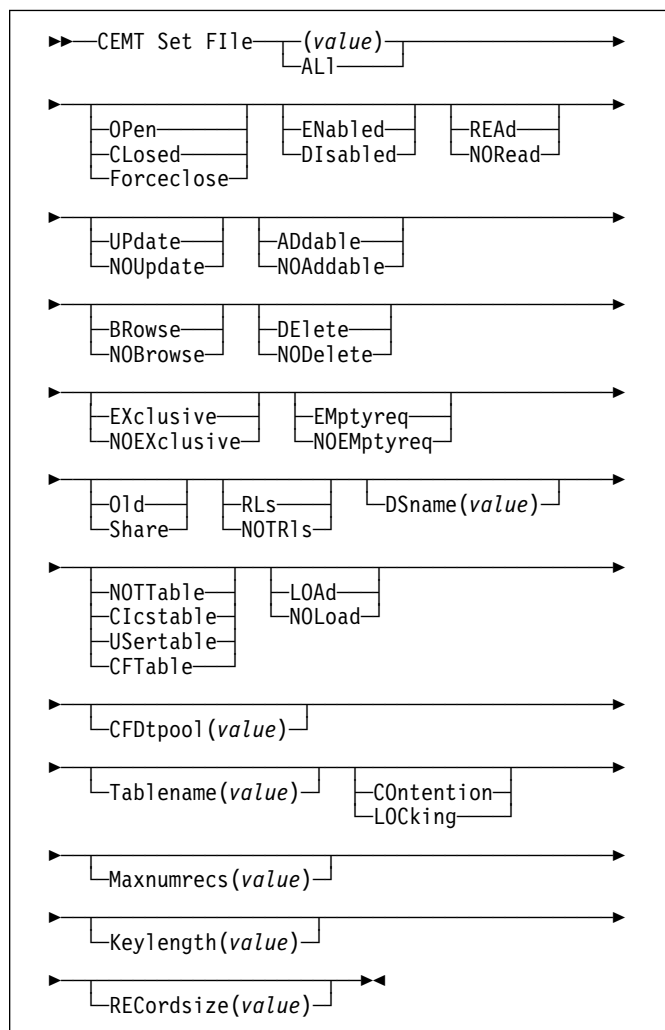
Options

(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.

DIsabled

The file is not available for use by transactions except for those that are currently using it. If there are any such users, 'BEING DISABLED' is also displayed. The CEMT SET FILE DISABLED command has no effect on existing users; it simply prevents new users from accessing the file.

The file can be reenabled by a SET FILE ENABLED command. (See also UNENABLED.)

It is not possible to disable a remote file.

DSname**(**value**)**

specifies the name (1–44 characters) of the VSAM or BDAM data set with which this file is associated. CEMT SET FILE(value) DSNAME(value) associates the data set with the file and causes the FILECOUNT of the data set to be incremented by 1. You can use CEMT SET FILE DSNAME with a null DSNAME value to dissociate a file from its current data set.

Note: DSNAME replaced OBJECTNAME.

EMptyreq **(**vsam only**)**

Applies only to a VSAM file that has been defined to VSAM as a cluster with the REUSE parameter, and which is opened in non-RLS access mode. It indicates that when a file is next opened, its data is erased.

If the file is opened in RLS access mode this parameter has no effect.

A CEMT SET FILE EMPTYREC command can be applied only to files that are closed and either disabled or unenabled, and has no effect on the data until the file is opened. A SET FILE EMPTYREQ command can be canceled by a SET FILE NOEMPTYREQ command.

ENabled

The file is available for use by transactions and, if closed, it is opened on the first request. If the command CEMT SET FILE(value) CLOSED ENABLED is given while the file is in use, the status of the file becomes 'closed-unenabled'. If the command CEMT SET FILE(value) CLOSED is given while the file is in use, the file is disabled to prevent new users accessing the file.

EXclusive

Records on the file are protected from simultaneous update by being placed under exclusive control when a read for update is issued.

Forceclose

Similar to SET FILE CLOSED, except that any existing users of the file are abended (abend code 'AKC3' or 'ATCH'). As for CLOSED, the close is effected by the last of these users.

> ┌─ **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.

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

For information about switching from RLS to non-RLS mode, see the *CICS Recovery and Restart Guide*.

NOTTable
The file name does not represent a data table.

NOUpdate
You cannot update records in the file.

Old
The disposition of the data set associated with the file is defined as unshared for dynamic allocation.

OPen
The file is open. If the file is OPEN ENABLED, it is available for data accesses by CICS transactions. If it is OPEN DISABLED, it must first be enabled before it is available.

If the file was unenabled by a previous SET FILE CLOSED command, the SET FILE OPEN command reenables the file implicitly.

REAd
You can read records in the file.

RECordsize
specifies the maximum record size for a coupling facility data table that does not require pre-loading from a source data set.

You can specify the record size for a file that does not currently refer to a coupling facility data table, but which could be switched to use a coupling facility data table at a later date. Specify a record size of 1- to 32767-bytes, or specify a record size of zero to remove a previously defined value.

RLs **(**vsam only**)**
The file is defined to be accessed in RLS mode.

Share
The disposition of the data set associated with the file is defined as shared for dynamic allocation.

Tablename**(**value**)**
specifies the name of the table for a coupling facility data table.

UPdate
You can update records in the file.

USertable
   The file name represents a user-maintained data table.

CEMT SET GTFTRACE

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, cemt s g sta will cause CICS GTF tracing to be started.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

```
►►──CEMT Set Gtftrace──┬──────┬──►◄
                       ├─STArt─┤
                       └─STOp──┘
```
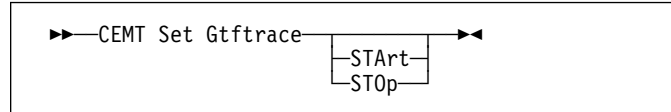
Options

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 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 overtype it with the required value.

- Type CEMT SET INTTRACE (CEMT S IN) followed by the attribute setting that you wish to change. For example, cemt s in sta will cause CICS internal tracing to be started.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

```
►►──CEMT Set INttrace──┬───────┬──►◄
                       ├─STArt─┤
                       └─STOp──┘
```
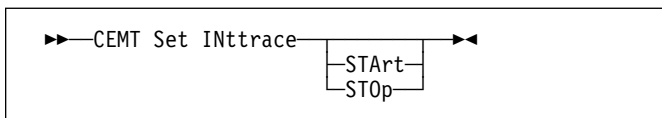
Options

STArt
    CICS internal tracing is started.

STOp
    CICS internal tracing is stopped.

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 overtype it with the required value.

- Type CEMT SET IRC (CEMT S IR) followed by the attribute setting that you wish to change. For example, cemt s ir o will cause interregion communication to be initialized.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

```
►►──CEMT SET IRc──┬─Open─────┬──►◄
                  ├─Closed───┤
                  └─Immclose─┘
```
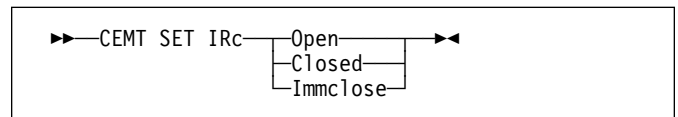
Options

Closed
    Interregion communication for this CICS system is to be terminated after all tasks using IRC sessions have terminated normally.

Immclose
    All tasks using IRC sessions are to be abended when they next try to use the session. Interregion communication is terminated when all tasks that were using IRC have been abended or have terminated normally.

Open
    Interregion communication for this CICS system is to be initialized. This allows connections to be established between this system and other systems, including DL/I batch regions and non-CICS client programs using the external CICS interface (EXCI). You can control the availability of individual connections with the INSERVICE|OUTSERVICE settings on the connection names.

    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

```
►►──CEMT Set Journalname(value)──┬──(value)──┬──►
                                 └──All──────┘

►──┬────────────┬──┬────────┬──►◄
   ├─Enabled────┤  ├─FLush──┤
   └─DIsabled───┘  └─Reset──┘
```
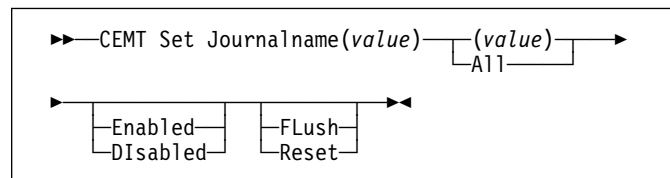
Options

(value**)**
Specifies the journal name.

To modify journals defined with a numeric identifier in the range 1–99, specify journal name DFHJ*nn*, where *nn* is the journal number.

You cannot specify DFHLOG or DFHSHUNT, because you are not allowed to modify the status of the system log.

All specifies that any changes you request are made to all resources of the specified type that you are authorized to access.

DIsabled
The journal is disabled. It cannot be used again until it is reenabled by a SET JOURNALNAME(...) STATUS(ENABLED) or ACTION(RESET) command.

Enabled
The journal is open and is available for use.

FLush
The log buffers are written out to the log stream but the journal is not closed.

You can use this option to ensure that all current records are written out to the log stream before the stream is processed using a batch utility.

Reset
The journal is disabled, but can be reenabled by the next SET JOURNALNAME STATUS(ENABLED) or ACTION(RESET) command for this journal.

CEMT SET 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 49), but you cannot use the symbols * and + to specify a family of identifiers.

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

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

- Type CEMT SET LINE (CEMT S L) with a termiinal identifier, followed by one or more attribute settings that you wish to change. For example, cemt s l(*value*) i will make the line available for use (inservice).

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

```
►►──CEMT Set Line(value)─────────────────────►
                       └─NUmber(value)─┘

►──────┬─────────────────┬──┬─Inservice──┬────►
       └─NEgpoll(value)──┘  └─Outservice─┘

►──┬────────────┬──►◄
   ├─Purge──────┤
   └─Forcepurge─┘
```
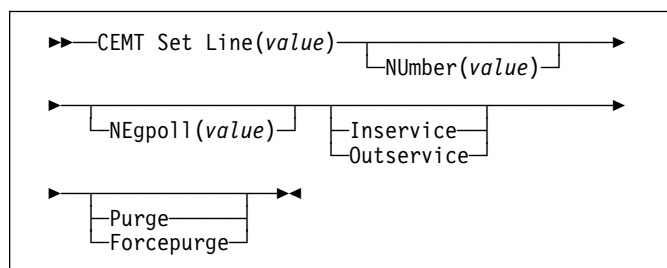
Options

(*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.

NEgpoll**(***value***)**
   specifies the negative poll delay interval (00000–20000 milliseconds). For a TCAM line, this specifies the interval before control is passed to the terminal error program (DFHTEP) when a CICS task is not ready to accept a record from an input process queue.

   If the CICS task issues a read before the interval expires, processing continues normally and DFHTEP is not notified. The default value is zero.

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

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 overtype them with the required values.

- Type `CEMT SET MODENAME` (`CEMT S MOD`) with either a modename or ALL, followed by one or more attribute settings that you wish to change. For example, `cemt s mod(modename) acq` will cause CICS to acquire all the negotiated contention-winner sessions within the group named by modename.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

```
►►──CEMT Set MODename──┬──(value)──┬──────────────────►
                       └──ALl──────┘

►──┬───────────────────────┬────────────────────────►
   └──COnnection(value)─────┘

►──┬───────────────────┬──┬──ACQuired──┬──►◄
   └──AVailable(value)──┘  └──CLosed────┘
```
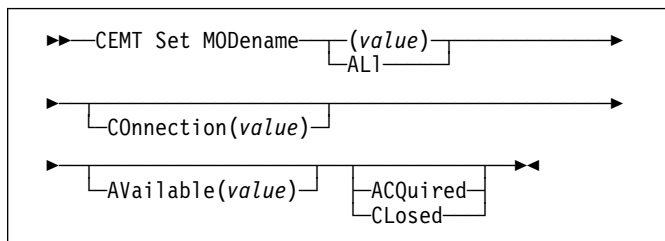
Options

(value**)**
  specifies one or more names (1-8 characters) of a group of sessions. Mode names need not be unique. You cannot specify a list of names, but you can use the symbols * and + to specify a family of mode names.

  A mode name consisting of eight blanks is valid. To specify this name, enclose the string of blanks within single quotation marks:

  `MODENAME('        ')`

ACQuired
  CICS is to acquire (or "bind") all the negotiated contention-winner sessions within the group.

ALl
  specifies that any changes you request are done to all resources of the specified type that you are authorized to access.

AVailable**(**value**)**
  specifies the maximum number of sessions within the group that can be concurrently allocated for use. This value has to be the same for both sides involved in the session.

CLosed
  CICS is to release all sessions within the group. The sessions are quiesced and unbound. No further sessions can be acquired by another CICS system. No further activity is possible on the sessions within the group until the group is reacquired by the same CICS system in which it has been closed.

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

CEMT SET MONITOR

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:

      Sysevent

        Monitoring data is not written at transaction termination.

      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.

   c. Exception 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.

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

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

- Type `CEMT SET MONITOR` (`CEMT S MON`) followed by one or more attribute settings that you wish to change. For example, `cemt s mon on p ex` will set CICS monitoring ON and collect performance-class and exception-class data.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

```
►►──CEMT Set MONitor──┬──────┬──┬────────┬──────────►
                      ├─ON──┤  ├─Perf───┤
                      └─OFf─┘  └─NOPerf─┘

►──┬──────────┬──┬──────────┬──────────────────────►
   ├─EVent───┤  ├─EXcept───┤
   └─NOEVent─┘  └─NOEXcept─┘

►──┬─────────────┬──┬──────────────┬───────────────►
   ├─Converse───┤  ├─SYncpoint────┤
   └─NOConverse─┘  └─NOSyncpoint──┘

►──┬──────────────────┬──►◄
   └─Frequency(value)─┘
```
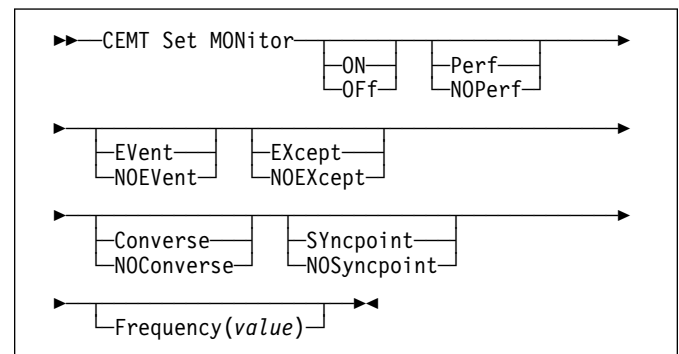
Options

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).

EVent

  The SYSEVENT class of monitoring data is to be collected.

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 001500–024000. 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).

NOEVent

The SYSEVENT class of monitoring data is not to be collected.

NOEXcept

The exception class of monitoring data is not to be collected.

NOPerf

The performance class of monitoring data is not to be collected.

NOSyncpoint

A transaction performance class record is not to be produced when a transaction takes an explicit or implicit syncpoint (unit-of-work), except when the syncpoint is part of task termination or a syncpoint rollback.

OFf

CICS monitoring is to be switched off. When monitoring is OFF, no monitoring data is accumulated or collected, irrespective of the settings of the monitoring data classes.

ON

CICS monitoring is to be switched on. When monitoring is ON, monitoring data is accumulated continuously, and is collected (written out to an SMF data set) for each of the classes of monitoring data that are active.

Perf

The performance class of monitoring data is to be collected.

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

Function: Change the status of networks.

Description: If you are familiar with network names, you may prefer to use this command as an alternative to CEMT SET TERMINAL (described under "CEMT SET TERMINAL" on page 208). However, CEMT SET NETNAME cannot be used for APPC sessions.

"netname" is the name by which the remote system or terminal is known to the network. The remaining operands for NETNAME are identical to those for TERMINAL.

Netnames need not be unique, and you cannot specify a list of names.

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

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

- Type `CEMT SET NETNAME` (`CEMT S AUTO`) with netname or ALL, followed by one or more attribute settings that you wish to change. For example, `cemt s n(netname) i at` will reset the named remote system or terminal to inservice and available for use.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.
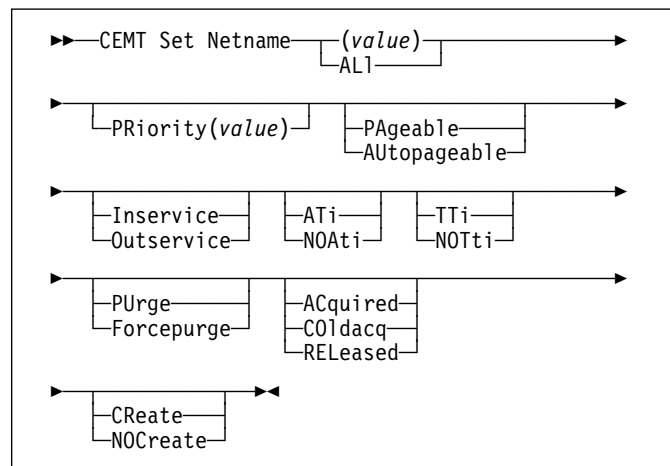
Options

(value**)**

specifies the name by which the remote terminal is known in the network.

ACquired (*vtam only*)
CICS is in session with the logical unit represented by the terminal.

ALl
specifies that any changes you request are made to all resources of the specified type that you are authorized to access.

ATi (*vtam only*)
The terminal is available for use by transactions that are initiated automatically from within CICS or, if the terminal is an ISC session, by transactions that are using this session as an alternative facility to communicate with another system.

AUtopageable (*vtam only*)
Pages after the first in a series are to be written to the terminal automatically.

COldacq (*vtam only*)
This is a special form of ACQUIRED, where no resynchronization is required. If the previous session abended, the use of COLDACQ overrides CICS integrity control. This could lead to integrity problems. Also, you should check the CSMT log for an activity keypoint after the restart of a session following a CICS failure. If there is no activity keypoint, you should issue COLDACQ again after the next emergency restart.

CReate (*vtam only*)
If the terminal is not in session, CICS is to acquire it if it is needed to satisfy an ATI request. This cannot be specified for IRC sessions.

Forcepurge
All transactions running with this terminal are immediately terminated abnormally. Data integrity is not guaranteed. In some extreme cases (for example, if an error occurs during backout processing), CICS might terminate abnormally.

Inservice (*vtam only*)
The terminal is available for use. For VTAM, INSERVICE means that the terminal can be ACQUIRED.

NOAti (*vtam only*)
The terminal is not available for use by transactions that are initiated automatically from within CICS or, if the terminal is an ISC session, by transactions that are using this session as an alternative facility to communicate with another system.

> **Note:** A terminal cannot be defined with both NOATI and NOTTI.

NOCreate (*vtam only*)
If the terminal is not in session, CICS is not to acquire it to satisfy an ATI request. A session must be started by, for example, a logon request or a CEMT SET TERMINAL ACQUIRED command before the ATI request can be satisfied.

If NOCREATE is set for an LU6.1 ISC session and there are no allocatable sessions left, the connection is placed OUTSERVICE.

NOTti (*vtam only*)
This terminal cannot be used by transactions.

> **Note:** A terminal cannot be defined with both NOATI and NOTTI.

Outservice (*vtam only*)
The terminal is not available for use. Setting a terminal OUTSERVICE means that the terminal can no longer be used by transactions. If PURGE or FORCEPURGE is also specified, any transaction using the terminal is terminated abnormally. If PURGE or FORCEPURGE is not specified, the transaction is allowed to terminate normally, but no further transactions are allowed to use the terminal. For VTAM, setting a terminal OUTSERVICE also causes it to be released and the operator to be signed off, either immediately or when the current transaction has terminated.

In an LU6.1 ISC session the connection is set OUTSERVICE if there are no allocatable sessions left.

To set an IRC session OUTSERVICE, you must set the connection to the MRO partner OUTSERVICE.

PAgeable (*vtam only*)
Pages after the first in a series are to be written to the terminal only if you request them to be.

PRiority(*value) (vtam only*) **(VTAM only)**
specifies the priority of a terminal relative to other terminals. The priority of a task is the sum of the transaction priority, the terminal priority, and the operator priority. Priority has no meaning for terminals that are ISC sessions being used as alternative facilities. The value must be in the range 0–255, where 255 is the highest priority.

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

RELeased (*vtam only*)
CICS is not in session with the logical unit represented by the terminal. Setting a terminal RELEASED causes the session to be terminated. Running transactions are allowed to finish unless PURGE or FORCEPURGE is also specified.

TTi (*vtam only*)
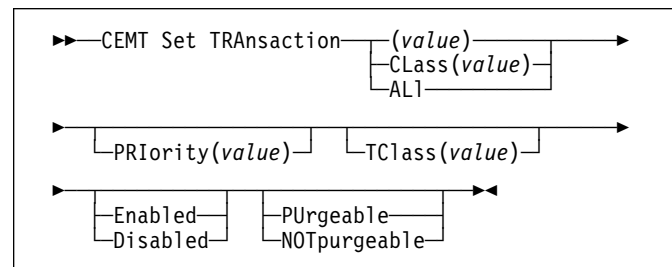This terminal can be used by transactions.

CEMT SET 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

```
               ┌─AL1─────┐
►►──CEMT Set PROCesstype─┤         ├───────────────►
               └─(value)─┘

►────┬──────────┬──┬─Enabled──┬──►◄
     ├─ACtivity─┤  └─Disabled─┘
     ├─FUll─────┤
     ├─Off──────┤
     └─Process──┘
```
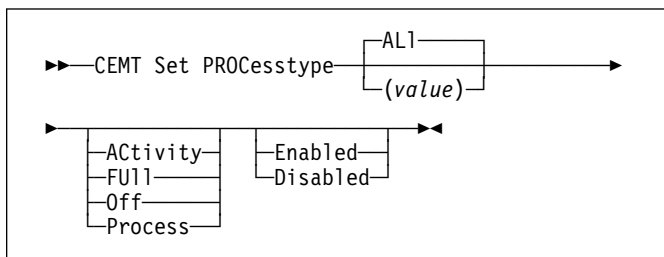
Options

ACtivity|FUll|Off|Process
specifies the level of audit logging to be applied to processes of this type.

Note: If the AUDITLOG attribute of the installed PROCESSTYPE definition is not set to the name of a CICS journal, an error is returned if you try to specify any value other than *OFF*.

The values are:

ACtivity
Activity-level auditing. Audit records will be written from:

1. The process audit points
2. The activity primary audit points.

FUll Full auditing. Audit records will be written from:

1. The process audit points
2. The activity primary *and* secondary audit points.

Off No audit trail records will be written.

Process
Process-level auditing. Audit records will be written from the process audit points only.

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

ALl
specifies that any changes you specify are made to all process-types that you are authorized to access.

Enabled Disabled
specifies whether new processes of this type can be created. The values are:

Disabled
The installed definition of the process-type is disabled. New processes of this type cannot be defined.

Enabled
The installed definition of the process-type is enabled. New processes of this type can be defined.

PROCesstype(value)
specifies the 8-character name of the process-type whose attributes are to be changed.

CEMT SET PROGRAM

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 PR`) with pgrmid, CLASS(*value*), or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE PROGRAM. You can then tab to the highlighted or blank fields and overtype them with the required values.

- Type `CEMT SET PROGRAM` (`CEMT S PR`) followed by one or more attribute settings that you wish to change. For example, `cemt s pr(pgrmid) e ne` will reset the values for the named program to make it available for use (enabled) and a new copy of the program will be used when all the transactions currently using the program have finished (newcopy).

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

```
►►──CEMT Set Program──┬──(value)───────┬──────►
                      ├─CLass(value)───┤
                      └─ALl────────────┘

                            PRIvate
►──┬──────────┬──┬─────────┬──┬─NEwcopy──┬──►
   ├─Enabled──┤  └─Shared──┘  └─PHasein──┘
   └─DIsabled─┘

►──┬───────┬──┬─DPlsubset─┬──────────────────►
   ├─CEDf──┤  └─Fullapi───┘
   └─NOCedf┘

►──┬─────────────────┬──┬─DEBUG───┬──────────►
   └─JVMClass(value)─┘  └─NODEBUG─┘

►──┬─HOtpool────┬──┬─JVM───┬──►◄
   └─NOTHotpool─┘  └─NOJVM─┘
```
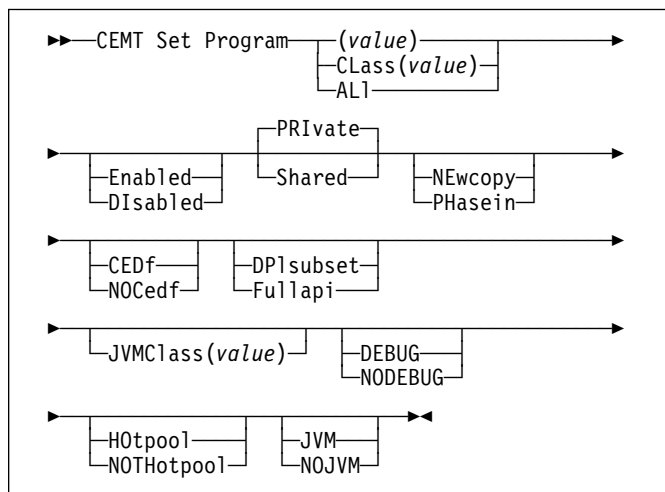
Options

(value**)**

specifies a program identifier (1–8 characters), which is the name of a specific program entry in the table of installed program definitions. If you omit the program name, the ALL option is assumed by default.

You cannot specify the CLASS option and a program name.

ALl

specifies that any changes you request are to be made to all resources of the specified type that you are authorized to access.

CEDf

When EDF is active, CEDF initiation and termination screens are to be shown by CEDF while this program is running. Other screens are also to be shown unless the program was translated using the NOEDF translator option.

CLass**(***value***)**

specifies the 2-character suffix of a program list table (PLT). Use this option to inquire on those programs defined in the specified PLT. Note that, unlike a shutdown PLT, it is not mandatory to define a program definition for a startup PLT. However, if you want to use the CLASS option, you must define and install a program resource definition for the PLT specified by "clasid".

If the clasid specified is the suffix of a PLT that does not have a program definition, the INQUIRE PROGRAM CLASS(clasid) command returns a 'CLASS NOT FOUND' message.

You cannot specify CLASS and a program name.

DEbug

The JVM is to operate in debugging mode for this program.

DIsabled

The program is not available for use. Programs beginning with "DFH" cannot be disabled because these characters are reserved for use by CICS.

DPlsubset

The program is restricted to the DPL API subset, as for a distributed program link request, when it runs in the local CICS region.

A program is always restricted to the DPL subset when it is invoked in a remote region via a DPL request, regardless of this option.

Enabled

The program is available for use.

Fullapi

The program is not restricted to the DPL subset of the CICS API when it runs in the local CICS region, and can use the full API.

Hotpool

The Java program object is to be run in a preinitialized Language Environment enclave.

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.

NEwcopy
CICS is to use 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.

> **Note:** NEWCOPY cannot be specified either for a program specified with the HOLD option. This value is ignored for a JVM program.

NODebug
The JVM is not to operate in debug mode for this program.

NOJvm
The program is not to run under the control a 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.

NOTHotpool
The Java program object is not to be run in a preinitialized Language Environment enclave.

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. This value is ignored 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

Comments: See "CEMT SET TDQUEUE" on page 207. (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.
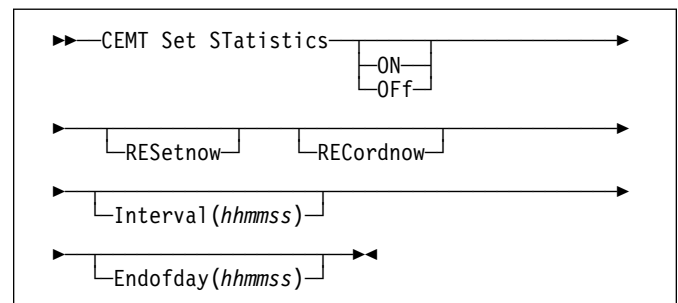
Options

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.

```
┌'─ CICS initialized with
│   ENDOFDAY(170000)
│   INTERVAL(030000)
│
│
│
│
│
│
▼
'─''''─''''─''''─''''─''''─''''─''''─''''─''''─
 T    T    T    T    T    T    T    T    T    T
 08   09   10   11   12   13   14   15   16   17
           I              I              E
```

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.
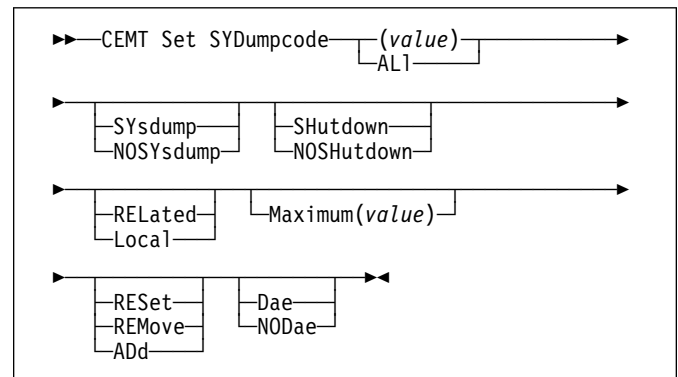
Options

(*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'.

CEMT SET SYDUMPCODE

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 could lead to dump suppression even though NODAE is set here. For information about DAE, SUPPRESS, and SUPPRESSALL, see the *OS/390 MVS Diagnosis: Tools and Service Aids* manula.

NOSHutdown
> CICS is not to be shut down after the occurrence of an error corresponding to this dump code.

NOSYsdump
> A system dump is not to be taken for this system dump code.

RELated
> A system dump request is to be sent to MVS images in the sysplex which are running XCF/MRO connected CICS regions related to the CICS region on which the dump is initiated.

> **Note:** A related CICS region is one in which the unit of work identifiers, in the form of APPC tokens, of one or more tasks match those in the CICS region that issued the dump request.

REMove
> This system dump code is removed from the system dump code table.

RESet
> The current number of calls for this dump code is reset to zero.

SHutdown
> CICS is to be shut down.

SYsdump
> A system dump is to be taken for this system dump code.

CEMT SET SYSTEM

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 overtype them with the required values.

- Type `CEMT SET` followed by one or more attribute settings that you wish to change. For example, `cemt s ag(value) ma(value)` resets the values for priority aging and the maximum number of tasks allowed at any one time in the CICS system.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

```
►►─CEMT Set SYStem─┬─────────────────────────┬─►◄
                   ├─AGing(value)────────────┤
                   ├─AKp(value)──────────────┤
                   ├─DSAlimit(value)─────────┤
                   ├─DSRtprogram(value)──────┤
                   ├─DTrprogram(value)───────┤
                   ├─EDsalimit(value)────────┤
                   ├─Forceqr─┬─Force───┬──────┤
                   │         └─NOForce─┘      │
                   ├─Logdefer(value)─────────┤
                   ├─MAXOpentcbs(value)──────┤
                   ├─MAxtasks(value)─────────┤
                   ├─MRobatch(value)─────────┤
                   ├───┬─CTLGAll────┬─────────┤
                   │   ├─CTLGModify─┤         │
                   │   └─CTLGNone───┘         │
                   ├─Progautoexit(value)─────┤
                   ├─┬─AUTOActive───┬─────────┤
                   │ └─AUTOInactive─┘         │
                   ├─RUnaway(value)──────────┤
                   ├─SCandelay(value)────────┤
                   ├─Time(value)─────────────┤
                   └───┬─SYsdump───┬──────────┘
                       └─NOSysdump─┘
```
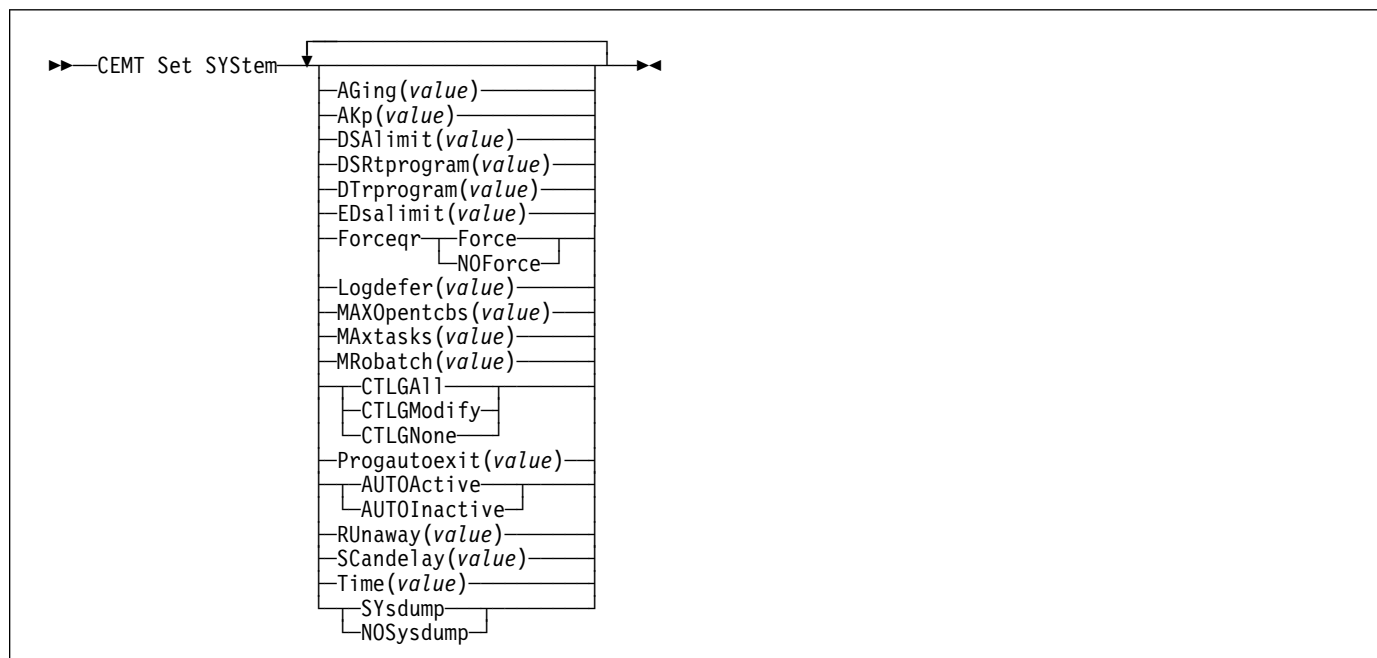
Options

AGing**(**value**)**
    specifies the time factor for CICS to use in the priority aging algorithm for incrementing the priority of a task. The value is expressed as "milliseconds per unit of priority", and must be in the range 00000–65535. The priority aging factor is defined initially by the PRTYAGE system initialization parameter, in which the value represents the number of milliseconds that must elapse before the priority of a waiting task can be adjusted upward by 1. See the *CICS System Definition Guide* for information about the PRTYAGE parameter.

AKp**(**value**)**
    specifies the activity keypoint trigger value, which is the number of write requests to the CICS system log stream output buffer between the taking of keypoints. It must have a value in the range 00200–65535. You cannot change this value if AKPFREQ=0 was specified at system initialization.

AUTOActive
    Autoinstall for programs is active. On first use, if a program, mapset, or partitionset is not defined, the definition is created dynamically.

AUTOInactive
    Autoinstall is not active. If a program is not defined, a PGMIDERR or transaction abend occurs when it is referenced.

CTLGAll
    All autoinstalled program definitions are to be cataloged and restored on a warm or emergency restart.

CTLGModify
    Autoinstalled program definitions are to be cataloged only if they are modified (for example, by a CEMT SET PROGRAM command), so that the modified definitions are restored on a warm or emergency restart.

CTLGNone
    No autoinstalled program definitions are to be cataloged. They are autoinstalled again after a warm or emergency restart.

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.

| **DSR**tprogram**(**_value_**)**
| specifies the name of the distributed routing program.

| **DT**rprogram**(**_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.

| **F**orce
| Force all user application programs to run under the QR
| TCB, even if they are defined with the
| CONCURRENY(THREADSAFE) attribute.

| **L**ogdefer**(**_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.

| **MAXO**pentcbs**(**_value_**)**
| specifies the maximum number of open TCBs that can
| exist concurrently in the CICS region. The value
| specified can be in the range 1 to 999. If you reduce
| MAXOPENTCBS from its previously defined value, and
| the new value is less than the number of open TCBs
| currently allocated, CICS detaches TCBs to achieve the
| new limit only when they are freed by user tasks.
| Transactions are not abended to allow TCBs to be
| detached to achieve the new limit. If there are tasks
| queued waiting for an open TCB, and you increase
| MAXOPENTCBS from its previously defined value, they
| will be resumed when CICS attaches new TCBs up to
| the new limit.

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.

| **NOF**orce
| 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

Function: Change the status of tasks.

Description: You cannot specify a list of identifiers, nor can you use the symbols * and + to specify a family of tasks.

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

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

- Type CEMT SET TASK (CEMT S TA) with number or ALL, followed by one or more attribute settings that you wish to change. For example, cemt s ta(*value*) pu will make the specified task purgeable in system-stall conditions.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.
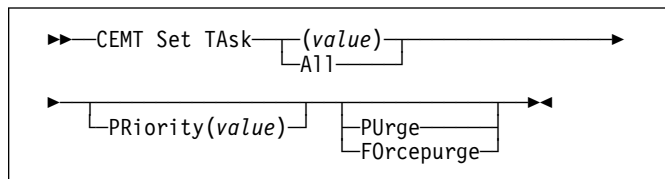
Options

(*value*)

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

All Any changes you request are made to all resources of the specified type that you are authorized to access.

FOrcepurge

The task is 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.

\# If the task is in any way associated with an open TCB
\# (for example, it involves a Java program running on a J8
\# or H8 mode TCB), you could experience a delay of
\# about a minute before the task is finally purged.

PRiority**(*value*)**

specifies the priority of the task, in the range 0–255 where 255 is the highest priority.

PUrge

The task is terminated. Task termination occurs only when 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 on a J8
\#   or H8 mode TCB), you could experience a delay of
\#   about a minute before the task is finally purged.

CEMT SET TCLASS

Function: Reset the maximum number of tasks and the purge threshold for a transaction class.

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

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

- Type CEMT SET TCLASS (CEMT S TC) followed by one or more attribute settings that you wish to change. For example, cemt s tc(*value*) p(3) will reset the purge threshold value to 3.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

```
►►──CEMT Set TClass──┬─(value)─┬──────────────────►
                     └─ALl────┘

►──────────┬──────────────────────┬─────────────────►
           └─Maxactive(value)─┘

►──────────┬──────────────────────┬──►◄
           └─Purgethresh(value)─┘
```
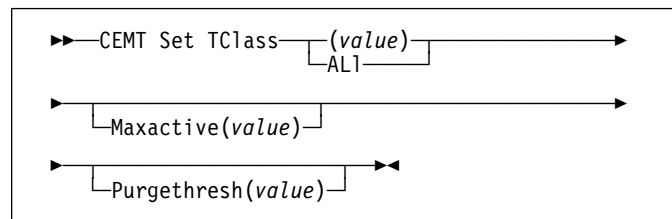
Options

(value**)**
 specifies the 8-character transaction class name. If no transaction class name is assigned, it defaults to DFHTCL00.

ALl
 specifies that any changes you request are made to all resources of the specified type that you are authorized to access.

Maxactive**(**value**)**
 specifies the largest number of transactions in the transaction class which are allowed to run concurrently. The value can be in the range 0-999.

Purgethresh**(**value**)**
 specifies the limit at which the queuing transactions are purged for the transaction class. It can have a value in the range 0–1000000. Zero means that the transactions are not purged. A nonzero number "n" means that the first (n-1) transactions to arrive are queued and the nth is purged.

CEMT SET TCPIP

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 overtype them with the required values.

- Type CEMT SET TCPIP (CEMT S TCPIP) with (*value*) or ALL, followed by one or more attribute settings that you wish to change. For example, cemt s tcpip *status* open opens CICS internal sockets support.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

```
►►──CEMT Set TCPIP──┬─(value)─┬──────────►
                    └─ALl────┘

 ►──┬─OPEN─────┬──►◄
    ├─CLOSE────┤
    └─IMMCLOSE─┘
```
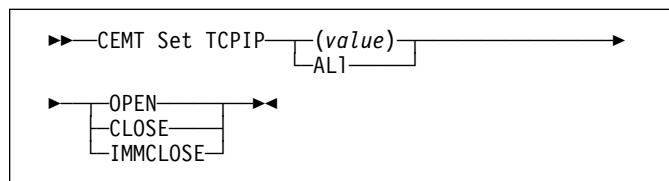
Options

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.

CEMT SET TCPIPSERVICE

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

Description: SET TCPIPSERVICE allows you to update the information relating to a service which uses CICS internal TCP/IP support.

Note: This command has no effect on the sockets support provided by the TCP/IP for 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 TCPIPSERVICE (the minimum abbreviation is CEMT S TCPIPS). You get a display that lists the current status, similar to that obtained by CEMT INQUIRE TCPIPSERVICE. You can then tab to the highlighted or blank fields and overtype them with the required values.

- Type CEMT SET TCPIPS (CEMT S TCPIPS) with (*value*) or ALL, followed by one or more attribute settings that you wish to change. For example, cemt s tcpips *status* open opens CICS internal sockets support.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

```
►►──CEMT Set TCPIPSERVICE───(value)──────────────►
                            └─ALl──┘

►──┬─────────────────┬──┬─OPEN─────┬──────────────►
   └─Backlog(value)──┘  ├─CLOSE────┤
                        └─IMMCLOSE─┘

►──┬────────────┬─►◄
   └─URM(value)─┘
```
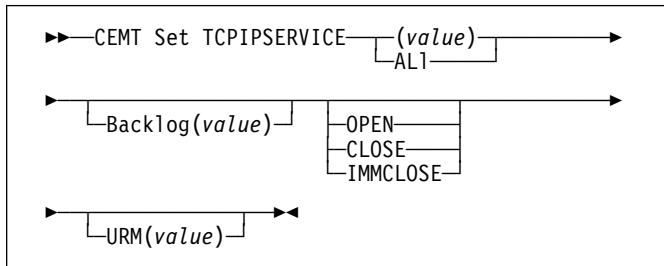
Options

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.

Openstatus
Changes the status of the service. CVDA values are:

OPEN CICS internal sockets support is to start listening for incoming requests on the specified port.

CLOSE CICS internal 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 can 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 overtype them with the required values.

- Type `CEMT SET TDQUEUE` (`CEMT S TD`) with (*value*) or ALL, followed by one or more attribute settings that you wish to change. For example, `cemt s td(value) en op` will make a named extrapartition queue accessible by applications and open.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.
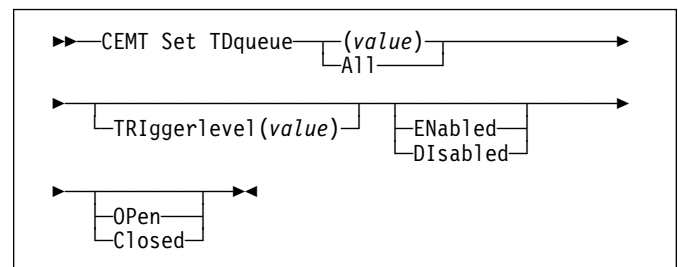
Options

(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 or ACF/TCAM GSA 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 192.

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

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

- Type `CEMT SET TERMINAL` (the minimum abbreviation is `CEMT S TE`) with (*value*), CLASS(*value*), or ALL. You get a display that lists the current status, similar to that obtained by CEMT INQUIRE TERMINAL. You can then tab to the highlighted or blank fields and overtype them with the required values.
- Type `CEMT SET TERMINAL` (`CEMT S TE`) with (*value*), CLASS(*value*), or ALL, followed by one or more attribute settings that you wish to change. For example, `cemt s te(value) i cr` specifies that a named terminal is available for use (inservice) and that it can be acquired automatically by ATI transactions (create).

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

```
►►──CEMT Set TErminal────(value)─────────────►
                       ├─CLass(value)─┤
                       └─ALl──────────┘

►──────────────────────────────────────────►
      └─PRiority(value)─┘  ┌─PAgeable─────┐
                          └─AUtopageable─┘

►──────────────────────────────────────────►
   ┌─Inservice──┐  ┌─ATi───┐  ┌─TTi───┐
   └─Outservice─┘  └─NOAti─┘  └─NOTti─┘

►──────────────────────────────────────────►
   ┌─PUrge──────┐  ┌─ACquired─┐
   ├─Forcepurge─┤  ├─COldacq──┤
   └─CAncel─────┘  └─RELeased─┘

►──────────────────────────►◄
   ┌─CReate───┐
   └─NOCreate─┘
```
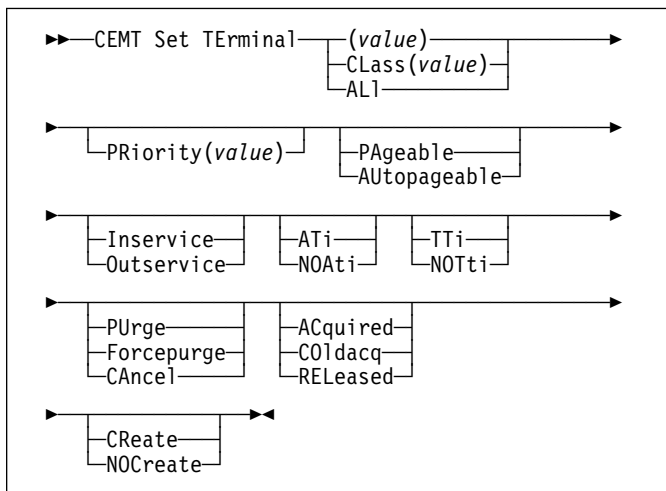
Options

(value**)**

A terminal identifier (1-4 characters) as specified in an installed terminal definition.

If the terminal name is, for example, S201, this option is coded thus:

```
CEMT SET TERMINAL(S201)
```

ACquired **(**vtam only**)**

CICS is in session with the logical unit represented by the terminal.

ALl

Any changes you request are made to all resources of the specified type that you are authorized to access.

ATi **(**vtam only**)**

The terminal is available for use by transactions that are initiated automatically from within CICS or, if the terminal is an ISC session, by transactions that are using this session as an alternative facility to communicate with another system.

AUtopageable **(**vtam only**)**

Pages after the first in a series are to be written to the terminal automatically.

CAncel

AIDs queuing for the specified terminal are canceled. AIDs representing scheduled and allocated requests waiting in the local CICS system for the specified terminal are canceled. However, TD AIDs with an associated triggered task already started are not canceled.

When a canceled scheduled request is found to have a precursor in a remote CICS system, this remote AID is canceled asynchronously. Message DFHTF0100 is written to CSMT to indicate how many AIDs have been deleted for the terminal and how many remain.

An 'AIDS CANCELED' message appears on the CEMT panel whenever AIDs are deleted using the CANCEL option of the CEMT SET TERMINAL command.

CLass**(**value**)**

The 1-or 2-character suffix of a terminal list table (TLT).

COldacq **(**vtam only**)**

This is a special form of ACQUIRED, where no resynchronization is required. If the previous session abended, the use of COLDACQ overrides CICS integrity control. This could lead to integrity problems. Also, you should check the CSMT log for an activity keypoint after the restart of a session following a CICS failure. If there is no activity keypoint, you should issue COLDACQ again after the next emergency restart.

CReate **(**vtam only**)**

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

Forcepurge

Any transaction running with this terminal is immediately terminated abnormally. Data integrity is not guaranteed. In some extreme cases (for example, if an error occurs during backout processing), CICS might terminate abnormally.

Inservice

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

NOAti **(**vtam only**)**

The terminal is not available for use by transactions that are initiated automatically from within CICS or, if the terminal is an ISC session, by transactions that are using this session as an alternative facility to communicate with another system.

Note: A terminal cannot be defined with both NOATI and NOTTI.

NOCreate **(**vtam only**)**

If the terminal is not in session, CICS is not to acquire it to satisfy an ATI request. A session must be started by, for example, a logon request or a CEMT SET TERMINAL ACQUIRED command before the ATI request can be satisfied.

If NOCREATE is set for an LU6.1 ISC session and there are no allocatable sessions left, the connection is placed OUTSERVICE.

NOTti **(**vtam only**)**

This terminal cannot be used by transactions that are initiated from this terminal.

Note: A terminal cannot be defined with both NOATI and NOTTI.

Outservice

The terminal is not available for use. Setting a terminal OUTSERVICE means that the terminal can no longer be used by transactions. If PURGE or FORCEPURGE is also specified, any transaction using the terminal is terminated abnormally. If PURGE or FORCEPURGE is not specified, the transaction is allowed to terminate normally, but no further transactions are allowed to use

the terminal. For VTAM, setting a terminal
OUTSERVICE also causes it to be released and the
operator to be signed off, either immediately or when the
current transaction has terminated.

In an LU6.1 ISC session the connection is set
OUTSERVICE if there are no allocatable sessions left.

To set an IRC session OUTSERVICE, you must set the
connection to the MRO partner OUTSERVICE.

PAgeable **(***vtam only***)**

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

PRiority**(***value) (vtam only***)**

specifies the priority of a terminal relative to other
terminals. The priority of a task is the sum of the
transaction priority, the terminal priority, and the operator
priority. Priority has no meaning for terminals that are
ISC sessions being used as alternative facilities. The
value must be in the range 0–255, where 255 is the
highest priority.

PUrge

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

RELeased **(***vtam only***)**

CICS is not in session with the logical unit represented
by the terminal. Setting a terminal RELEASED causes
the session to be terminated. Running transactions are
allowed to finish unless PURGE or FORCEPURGE is
also specified.

TTi **(***vtam only***)**

This terminal can be used by transactions.

CEMT SET TRANSACTION

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
  overtype them with the required values.

- Type `CEMT SET TRANSACTION` (`CEMT S TRANS`) followed by
  one or more attribute settings that you wish to change.
  For example, `cemt s trans(`*value*`) e pu` specifies that a
  named transaction is available for use (enabled) and is
  system-purgeable.

Typing ? at the beginning of either the first or second line
gives a syntax prompt. Resetting the values takes effect
immediately.

```
►►──CEMT Set TRANsaction──┬──(value)──────┬──────────────►
                          ├─CLass(value)──┤
                          └─ALl───────────┘

►──────┬──────────────────┬────┬───────────────┬───────────►
       └─PRIority(value)───┘    └─TClass(value)─┘

►──────┬──────────────┬─┬─────────────────┬──►◄
       ├─Enabled──────┤ ├─PUrgeable───────┤
       └─Disabled─────┘ └─NOTpurgeable────┘
```

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.
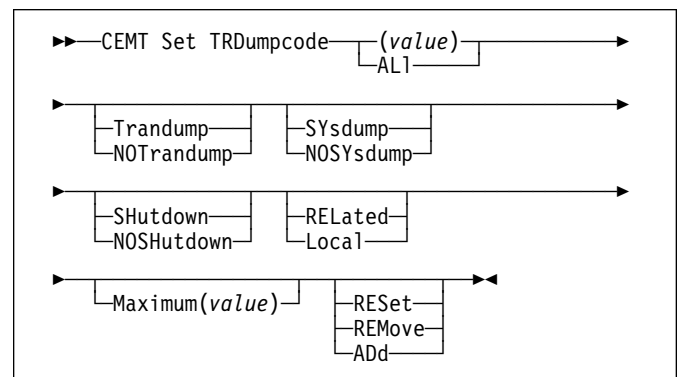
Options

(*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.

ALl
   specifies that any changes you request are made to all resources of the specified type that you are authorized to access.

Local

A system dump request is not to be sent to other MVS images in the sysplex which are running XCF/MRO connected CICS regions related to the CICS region on which the dump is initiated.

> **Note:** A related CICS region is one in which the unit of work identifiers, in the form of APPC tokens, of one or more tasks match those in the CICS region that issued the dump request.

Maximum**(**value**)**

specifies the largest number of dump calls for this dump code that result in dumps being taken. The range is 0–999, though a value of 999 means 'no limit'.

NOSHutdown

The CICS system is not to shut down.

NOSYsdump

A system dump is not to be taken.

NOTrandump

A transaction dump is not to be taken.

RELated

A system dump request is to be sent to MVS images in the sysplex which are running XCF/MRO connected CICS regions related to the CICS region on which the dump is initiated.

> **Note:** A related CICS region is one in which the unit of work identifiers, in the form of APPC tokens, of one or more tasks match those in the CICS region that issued the dump request.

REMove

This code is to be removed from the dump code table.

RESet

The current count of dump calls for this dump code is to be reset to zero.

SHutdown

The CICS system is to shut down after the occurrence of an error corresponding to this dump code.

SYsdump

A system dump is to be taken.

Trandump

A transaction dump is to be taken.

CEMT SET TSQUEUE

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 can be used to ensure that the queue to be deleted has not been referenced since a previous INQUIRE was issued. It can also be used to delete queues which have not been referenced within a given interval.

Syntax

```
>>──CEMT Set TSqueue──┬─────AL1─────┬──────────────>
                      └──(value)──┘

>──────┬──────────────────┬────────────────────────>
       ├─SYSID(value)────┤
       └─POOLNAME(value)─┘

>──────┬────────────────────┬──────────────────────>
       └─LAstusedint(value)─┘
 >──┬─Action(value)─┬──><
    └─DELETE────────┘
```
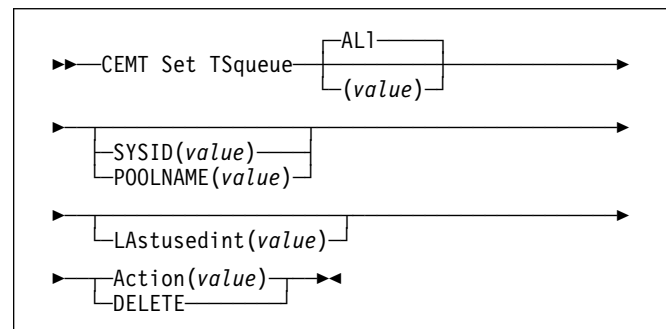
Options

(value**)**

specifies the identifier of the TS queue

ALl

specifies that any changes you request are made to all resources of the specified type that you are authorized to access.

ACTION(value**)**

specifies the action to be taken on the queue. The values are:

> DELETE
> the queue is to be deleted.

LASTUSEDINT

if this option is specified, the queue is only deleted if its last used interval is greater or equal to the value specified.

POOLNAME(data-value**)**

specifies an 8-character pool name.

SYSID(data-value**)**

specifies a 4-character shared sysid.

| **TSQUEUE(**data-value**)**
| specifies the 16-character identifier of the TS queue.

CEMT SET UOW

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

```
►►──CEMT Set UOW(value)──┬─(value)─┬──────────────►
                         └─ALl─────┘

►──┬─COMmit──┬──►◄
   ├─Backout─┤
   └─Force───┘
```
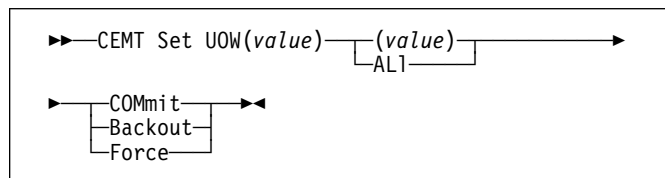
Options

(value**)**
      specifies the 16-character identifier of the UOW to be committed, backed out, or forced.

ALl
      specifies that any changes you request are made to all resources of the specified type that you are authorized to access.

Backout
      Attempt to force syncpoint backout processing, as specified for this UOW.

COMmit
      Attempt to force syncpoint commit processing, as specified for this UOW.

Force
      Attempt to force the UOW to backout or commit, as specified on the ACTION option of the TRANSACTION resource definition.

Note that all these values are valid only for UOWs that have been shunted in-doubt. For information about the in-doubt attributes of TRANSACTION definitions, see the *CICS Resource Definition Guide.*

CEMT SET UOWLINK

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 could have been created by connections that have since been discarded. If so, you could use the SET UOWLINK command to delete them. (For information about when it is safe to delete UOW-links, see the *CICS Intercommunication Guide*.)

Syntax

```
►►─CEMT Set UOWlink(value)──┬─(value)─┬──────►
                            └─ALl────┘

►─Delete─►◄
```
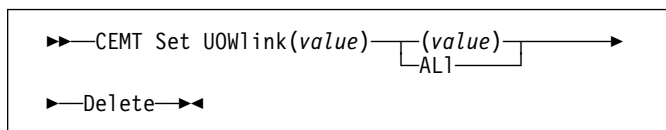
Options

(value**)**
   specifies the 4-character identifier of the UOW-connection dependency (the UOW-link) to which this command applies.

ALl
   specifies that any changes you request are made to all resources of the specified type that you are authorized to access.

Delete
   specifies that the UOW-link is to be deleted.

CEMT SET 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 overtype them with the required values.

- Type `CEMT SET VTAM` (`CEMT S VT`) followed by one or more attribute settings that you wish to change. For example, `cemt s vt o` specifies that the connection between CICS and VTAM is established.

Typing ? at the beginning of either the first or second line gives a syntax prompt. Resetting the values takes effect immediately.

```
►►─CEMT Set Vtam──┬─Open───────┬──────────►
                  ├─Closed─────┤
                  ├─Immclose───┤
                  └─Forceclose─┘

►─────┬──────────────────────┬────────────►
      └─Psdinterval(hhmmss)──┘

►─────┬───────────────┬─►◄
      └─Deregistered──┘
```
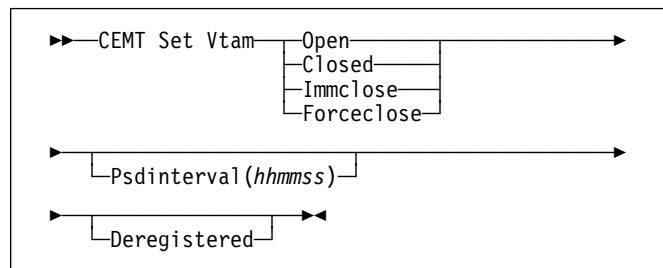
Options

Closed
   The connection between CICS and VTAM is to terminate normally; that is, transactions are allowed to complete and terminals are released normally.

Deregistered
   specifies that CICS is to be removed from the VTAM generic resource that it is currently a member of. If you deregister a region from membership of a generic resource, you should end any affinities that it owns—see the PERFORM ENDAFFINITY command.

   Generic resources are described in the *CICS Intercommunication Guide*.

Forceclose
   The connection is terminated immediately, and all terminal connections using the VTAM ACB (access method control block), together with the transactions associated with them, are terminated abnormally. If transactions do not terminate because of their state at the time FORCECLOSE was issued, it is necessary to issue CEMT SET TASK(nn) PURGE or FORCEPURGE before the ACB can be reopened.

   The difference between IMMCLOSE and FORCECLOSE is that IMMCLOSE cancels any task at a VTAM terminal

and then closes the terminal sessions tidily.
FORCECLOSE cancels any task and closes the VTAM
ACB, terminating all sessions immediately.

Immclose

The connection between CICS and VTAM is to be
terminated immediately. Any transactions using
VTAM-connected terminals are terminated abnormally,
and all such terminals are lost to CICS.

The difference between IMMCLOSE and FORCECLOSE
is that IMMCLOSE cancels any task at a VTAM terminal
and then closes the terminal sessions tidily.
FORCECLOSE cancels any task and closes the VTAM
ACB, terminating all sessions immediately.

Open

The connection between CICS and VTAM is established.
With ACF/VTAM, this refers to the opening of the
ACF/VTAM ACB through which ACF/VTAM
communicates with CICS.

A CEMT SET VTAM OPEN command will fail with a
STILL CLOSING message if the command is issued
while a previous predatory takeover is still being
processed. No attempt is made to open the VTAM ACB,
because it is still being closed as a result of the
predatory takeover. After the predatory takeover
completes, the CEMT SET VTAM OPEN command will
work as normal. The failure is notified by a
RESP=INVREQ and RESP2=14 condition.

If CICS is using VTAM multi-node persistent sessions,
and VTAM has been restarted after an abend, opening
the VTAM ACB causes CICS to restore the persistent
sessions that VTAM has retained. However, CICS does
not restore APPC synclevel 2 sessions, which are
unbound.

Psdinterval(*hhmmss*)

specifies the persistent session delay interval, expressed
in the form *hhmmss*. It specifies if and for how long
sessions are held in recovery-pending state after CICS
fails. The permitted range is 000000 to 235959 inclusive.

000000   If CICS fails, sessions are terminated.

Hhmmss   If CICS fails, sessions are held in
recovery-pending state by VTAM for up to
the interval specified.

Table 7 shows possible CEMT error messages resulting
from unsuccessful attempts to set the persistent session
delay interval.

Table 7. CEMT error messages for attempts to set the persistent session delay interval

Message	RESP/RESP2
INVALID PSDINT	RESP=INVREQ, RESP2=4
NOT WITH XRF	RESP=INVREQ, RESP2=8
SETLOGON FAILURE	RESP=INVREQ, RESP2=9
BACK LEVEL VTAM	RESP=INVREQ, RESP2=10
ACB CLOSED	RESP=INVREQ, RESP2=12
RECOVERY ERROR	All other cases of INVREQ

| # CEMT SET WEB

| **Function:**   Change Web garbage collection settings, or
| Web 3270 terminal timeout settings.

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

| • Type `CEMT SET WEB` (the minimum abbreviation is `CEMT S`
| `W`). You get a display that lists the current status, similar
| to that obtained by CEMT INQUIRE WEB. You can then
| tab to the highlighted or blank fields and overtype them
| with the required values.

| • Type `CEMT SET WEB` (`CEMT S W`) followed by one or more
| attribute settings that you wish to change. For example,
| `cemt s w g` specifies that you want to change the
| GARBAGEINT option.

| Typing ? at the beginning of either the first or second line
| gives a syntax prompt. Resetting the values takes effect
| immediately.

```
►►──CEMT Set Web──────────────────────────────►
                 └─Garbageint(value)─┘

►────────────────────────────►◄
    └─Timeoutint(value)─┘
```
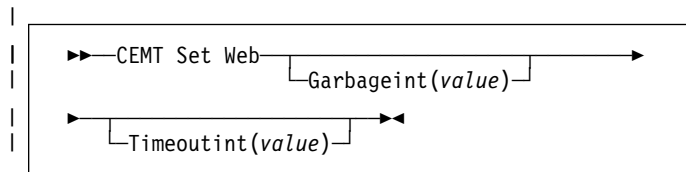
| **Displayed fields**

| **GARBAGEINT**
| is the interval at which the Web garbage collection task
| runs to clean up Web 3270 state data for which the
| terminal timeout interval has expired. The interval is
| measured in minutes.

| **TIMEOUTINT**
| is the period of time after which inactive Web 3270
| sessions are eligible for garbage collection. The period is
| measured in minutes.

Chapter 12. 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.

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.

CEOT in inquiry mode

Type CEOT. You get the following display:

```
  STATUS:  RESULTS - OVERTYPE TO MODIFY
  Ter(S208) Tra(CEOT) Pri(000) Pag Ins Ati Tti
     Net(IGKS208 ) Acq

  CEOT SYNTAX:
   < PAgeable | AUtopageable >
   < ATi | NOAti >
   < TTi | NOTti >
```

Figure 139. 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 overtype this value.

Aut

See 'AUTOPAGEABLE' in the Command Options below.

The paging status can be modified by overtyping.

ATi

See 'ATI' in the Command Options below.

Cre

This terminal can be acquired automatically by ATI transactions. You cannot overtype this value.

Ins

Your terminal is INSERVICE. You cannot overtype this value.

Net(*netname*)

displays the name by which this terminal is known to VTAM.

Pag

See 'PAGEABLE' in the Command Options below.

The paging status can be modified by overtyping.

Pri(*nnn*)

displays the priority of your terminal relative to other terminals, and is a number in the range 0–255.

Ter(*xxxx*)

displays the 4-character identifier of your terminal.

Tra(*ceot*)

displays the transaction that is running on your terminal. This is always CEOT.

Tti

The terminal is available for use by transactions initiated from this terminal. You cannot change the TTI setting by overtyping it with blanks.

CEOT in set mode

Type CEOT followed by one or more of the following attributes:

```
►►──CEOT──┬─────────────────┬──┬──────┬──────────►
          ├─PAgeable────────┤  ├─ATi──┤
          └─AUtopageable────┘  └─NOAti─┘

►──┬────────┬──►◄
   ├─TTi────┤
   └─NOTti──┘
```
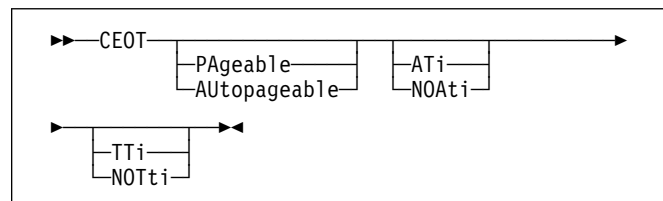
ATi

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.

CEOT—terminal status

NOTti
> This terminal cannot be used by transactions that are initiated from this terminal. Setting NOTTI is valid only for sequential devices.

PAgeable
> Pages after the first in a series are written to the terminal on request from the operator.

TTi
> This terminal can be used by transactions.

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 13. 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* manual.

The syntax of the CESF transaction is as follows:

```
►►──CESF──────────────────►◄
         ├─GOODNIGHT─┤
         └─LOGOFF────┘
```
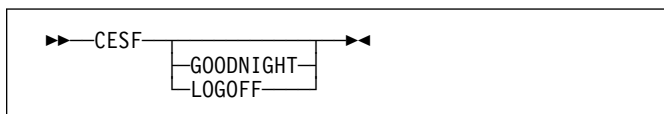
If you simply type CESF by itself, you are signed off the CICS system. Your terminal can still start transactions without you having to request them, and messages can be sent to it. You can sign on again at that terminal at any time.

The effects of typing CESF with the GOODNIGHT value or the LOGOFF value depend on the access method in use, as shown in Table 8:

Table 8. 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 14. 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 140).

```
 ******\  ******\  ******\   ******\         *\ ********\  ******\    ****\
 *******\  ******\  ******\  ******\       **\ ********\ ********\  ******\
 **\\\\**\   **\\\ **\\\**\ **\\\\**\     **\  **\\\\\\ **\\\\**\ **\\\\**\
 **\   \\  **\   **\  **\  \\ **\   \\    **\     **\   **\   \\ **\   **\
 **\        **\   **\  *******\     **\   **\     ******\  *******\ *******\
 **\        **\   **\  *******\ **\ **\   **\     ******\  *******\ *******\
 **\        **\   **\  \\\\**\  **\     **\ \\\\   **\\\\\\   \\\\\**\ **\\\\**\
 **\   **\  **\   **\ **\  **\  **\   **\      **\        **\   **\ **\   **\
 *******\  ******\  *******\  *******\ **\     ********\  *******\ **\   **\
 ******\\  ******\  ******\\  ******\\ *\     ********\  ******\\ **\   **\
  \\\\\\\    \\\\\\   \\\\\\    \\\\\\\  \       \\\\\\\\   \\\\\\  \\   \\
```
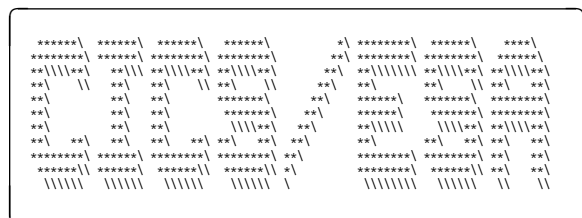
Figure 140. The CICS good-morning panel

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 141.

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 222.

2. For numeric-only devices or for the IBM 2741 Communication Terminal, use the signon procedure described in "Non-3270 family terminal" on page 222.

3. RACF recognizes only uppercase alphabetic characters. Userids and passwords specified in the CESN transaction are **only** translated to uppercase if the UCTRAN(YES) attribute is specified in the PROFILE resource definition or TYPETERM resource definition, otherwise data is passed to RACF as entered. See the *CICS Resource Definition Guide* for information on defining PROFILE and TYPETERM.

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
```
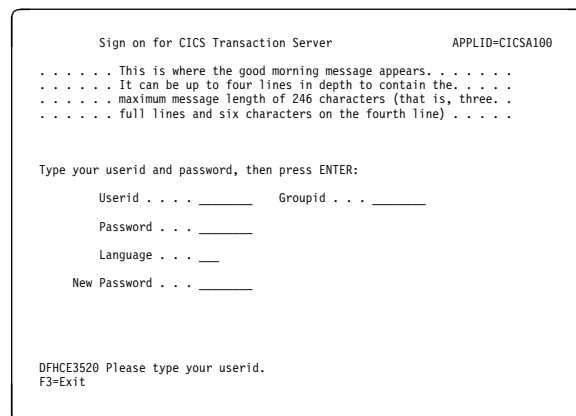
Figure 141. CESN transaction: initial screen

The panel for 40 x 12 size screens is shown below. There is no space for the release number or the good morning message.

```
      Sign on for CICS Transaction Server         APPLID=CICSA100
        Userid   . . . . _____
        Groupid  . . . . _____
        Password . . . . _____
        Language . . . . ___
        New Password . . _____



DFHCE3520 Please type your userid.
F3=Exit
```
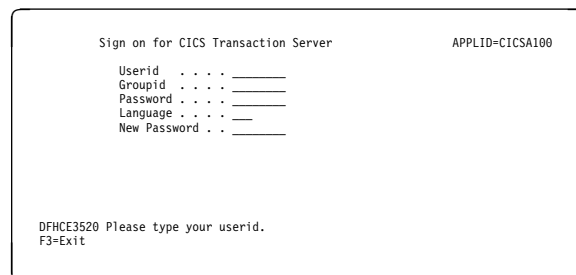
Figure 142. 40 x 12 screen

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:

```
|   ►►──CESN  USERID=uuuuuuuu──CESN──────────────►
|   ►──USERID=uuuuuuuu─────────────────────────────►
                       └─,GROUPID=gggggggg─┘
    ►───────────────────────────────────────────────►
         └─,LANGUAGE=lll─┘   └─,PS=pppppppp─┘
    ►─────────────────────►◄
         └─,NEWPS=password─┘
    Note:
    ¹ Do not use blanks between parameters in the
      comma-separated list.
```
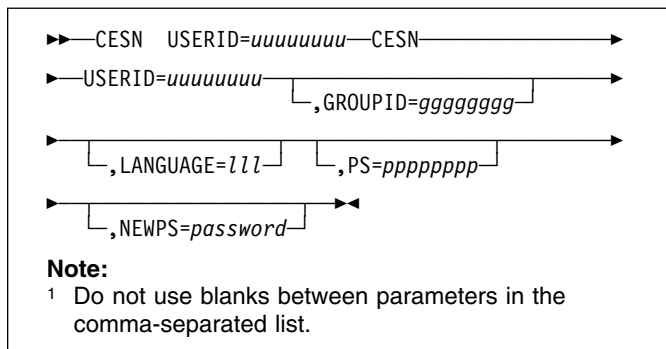
This style of input can be used on video display terminals, but is especially useful for sequential terminals and devices not belonging to the IBM 3270 family of terminals.

Command options

GROUPID=gggggggg
is your group userid (up to 8 characters). If this
parameter is specified, it may be propagated to remote
systems along with the userid (refer to the section on
'Implementing LU6.2 Security / Information about remote
users' or 'Implementing MRO security / Information about
remote users' in the *CICS RACF Security Guide*).

LANGUAGE=lll
enables you to choose a national language (from those that are available at your installation) for terminal directed messages. "lll", the national language identifier, is a standard, three-character IBM code. If the language identifier is omitted, CICS examines the following in this order:

1. Language specified for the user in the signon table

2. Language specified in the installed terminal definition for the terminal at which the signon occurs

3. Language specified on the NATLANG system initialization parameter.

If you enter a code other than the range of IBM codes, you get the following message:

`DFHCE3537 Language is invalid. Please retype.`

NEWPS=password
"password" is your new password (up to 8 characters). A 'password' value that consist entirely of space characters will not be interpreted as a password update request. The user will not be prompted for verification.

PS=pppppppp
is your password (up to 8 characters). If the password is required but omitted, then RACF will report a security

violation via message ICH408I before CICS prompts for the password.

USERID=uuuuuuuu
is your user ID (up to 8 characters).

Signing off with CESN

When a user starts the CESN transaction, CICS verifies userids and password by calling RACF. If the terminal user sign-on is valid, the CICS user domain keeps track of the signed-on user. Thereafter, CICS uses the information about the user when calling RACF to make authorization checks. If a user was signed on before the call to RACF, this signed-on user is signed off by CICS whether the returns from RACF are valid or invalid.

If users enter the CESN transaction when they are already signed on, the transaction may or may not sign them off. For example, a user is not signed off by entering CESN, and then pressing F3 when the signon panel is displayed.

In CICS Transaction Server for OS/390, 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 15. CEST—supervisory terminal

Use CEST to invoke a subset of the master terminal (CEMT) transaction. The CEST INQ|SET commands allows you to inquire about and alter some of the values of lines, netnames, tasks, and terminals.

CEST can be invoked from 3270 family display devices, or from equivalent terminals, or from the operating system console.

All of the CEST commands require the SUPRID keyword to be specified. SUPRID identifies the supervisor invoking the transaction, and its value is the 1-or 2-character suffix of a terminal list table (TLT). For information about the SUPRID keyword, see the *CICS Resource Definition Guide*.

For example, CEST INQ TERMINAL SUPRID(AL) displays the status of all the terminals in the DFHTLTA1 table. Values can be changed by overtyping the display or by using CEST SET commands. Apart from the requirement that SUPRID has to be specified, the syntax of the commands and the meaning of the values is exactly as described in CEMT INQUIRE|SET LINE, NETNAME, TASK, and TERMINAL.

Chapter 16. 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/ESA Trace Control Facility        CI41 IYAHZCCV

    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    9=Error List
```

Figure 143. 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 9. 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

Two other main functions of CETR are:

1. Setting the component trace options (press PF4)

2. Setting the transaction and terminal trace options (press
   PF5).

For information about the use of the various CETR options
as an aid to problem determination, see the *CICS Problem
Determination Guide*.

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 3
    Component Standard                      Special
    --------  ------------------------------  ------------------------------
       AP     1                               1-2
       BA     1                               1-2
       BF     1                               1
       BM     1                               1
       BR     1                               1-2
       CP     1                               1-2
       DC     1                               1
       DD     1                               1
       DI     1                               1
       DM     1                               1-2
       DS     1                               1-2
       DU     1                               1-2
       EI     1                               1
       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
```
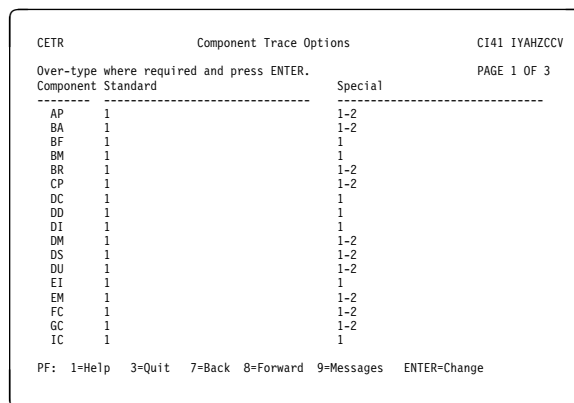
Figure 144. CETR transaction: component trace options screen

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.

CETR—trace control

The equal sign "=" can be used, when changing several fields, to indicate that a particular field should be changed to the same value as the one above.

The complete list of components is:

Component	Title
AP	Application domain
BA	CICS business application manager for BTS
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
DS	Dispatcher domain
DU	Dump domain
EI*	Exec interface
EM	Event manager domain for BTS
FC*	File control
GC	Global catalog domain
IC*	Interval control
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
PA	Parameter manager domain
PC*	Program control
PG	Program manager domain
RI*	Resource manager interface (RMI)
RM	Recovery manager domain
RX	Recoverable EXCI domain
SC*	Storage control macro-compatibility
SH	Scheduler services domain for BTS
SO	Sockets domain
SM	Storage manager 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

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.

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*.

Note: 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*.

Note: 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.

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 145. 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 144 on page 225, 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 145 on page 226, 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. For example, 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.

There are also fields that allow enabling of all nonterminal VTAM exits, and PF6 can be used to cancel all exit tracing.

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 146. CETR transaction: HELP panel (1)

```
CETR                Help: CICS Trace control facility

 (1) DATA ENTRY AND ALTERATION.

   This is achieved by simply over-typing the existing data, and then
   pressing ENTER.

   Abbreviations may be used, provided they are not ambiguous. The minimum
   possible abbreviations are listed in capitals to the right hand side of
   the input fields. For example STO may be entered for STOPPED, but ST
   would not be allowed as it could mean STARTED or STOPPED.

   Erased or blank fields will redisplay the current level settings as
   known to CICS. These will be the levels current when ENTER was last
   pressed.  They may have been changed from the previously displayed
   values by other activity in the system.

   All input will be validated when ENTER is pressed, and no internal
   settings will be made until all errors have been removed.


 PF 3=End  7=Back  8=Next  ENTER=End
```

Figure 147. CETR transaction: HELP panel (2)

```
CETR                Help: CICS Trace control facility

 (2) DESCRIPTION OF FIELDS.

 The values that can be set from the main panel are:-

   Internal Trace Status      -  Whether tracing to the main storage trace
                                 table is required.
   Internal Trace Table Size  -  The size of the main storage trace table.
   Auxiliary Trace Status     -  Whether tracing to the current auxiliary trace
                                 data set is required.
   Auxiliary Trace Dataset    -  The current data set. A = DFHAUXT, B = DFHBUXT.
   Auxiliary Switch Status    -  Whether a switch to the alternate extent is
                                 required at next or all end-of-extents.
   GTF Trace Status           -  Whether CICS is writing its trace data to GTF.
                                 Note that GTF must have been started to MVS.
   Master System Trace Flag   -  This flag must be on if CICS system-requested
                                 entries are to be written for STANDARD tracing
                                 tasks.
   Master User Trace Flag     -  This flag must be on if user-requested trace
                                 entries are to be produced.


 PF 3=End  7=Back  8=Next  ENTER=End
```

Figure 148. 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.
        PF9      Messages   Display any messages created, these can be
                            warning (informative) messages, or errors.




 PF 3=End  7=Back  ENTER=End
```

Figure 149. 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 150. 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 151. 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 152. CETR transaction: Component trace HELP panel (3)

```
CETR             Help: CICS Component Trace

 (3) MEANINGS OF COMPONENT ABBREVIATIONS.

 AP . . . Application domain      LD . . . Loader domain
 BF . . . Built-in function       LG . . . Log manager domain
 BM . . . Basic Mapping Support   LM . . . Lock manager
 CP . . . CPI-C interface         ME . . . Message domain
 DC . . . Dump compatibility layer MN . . . Monitoring domain
 DD . . . Directory manager       NQ . . . Enqueue Manager
 DI . . . Batch data interchange  PA . . . Parameter manager domain
 DM . . . Domain manager domain   PC . . . Program control
 DS . . . Dispatcher domain       PG . . . Program manager domain
 DU . . . Dump domain             RI . . . RMI
 EI . . . Exec interface          RM . . . Recovery Manager
 FC . . . File control            SC . . . Storage control
 GC . . . Global Catalog domain   SM . . . Storage Manager domain
 IC . . . Interval control        ST . . . Statistics domain
 IS . . . ISC                     SZ . . . Front End Prog Interface
 KC . . . Task control            TC . . . Terminal control
 KE . . . Kernel                  TD . . . Transient data
 LC . . . Local Catalog domain    TI . . . Timer domain

 PF 3=End  7=Back  8=Next  ENTER=End
```

Figure 153. CETR transaction: Component trace HELP panel (4)

```
CETR            Help: CICS Component Trace

 (4) MEANINGS OF COMPONENT ABBREVIATIONS.


 TR . . . Trace domain
 TS . . . Temporary storage
 UE . . . User exit interface
 US . . . User domain
 WB . . . Web interface
 XM . . . Transaction manager
 XS . . . Security domain











 PF 3=End  7=Back  8=Next  ENTER=End
```

Figure 154. 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 155. 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 156. 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 157. 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 158. 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 159. CETR transaction: Terminal and transaction HELP panel (4)

Chapter 17. 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.

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 231. 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.

Resource definitions

The required resource definitions for CIND are provided in group DFHINDT, which is a member of group list DFHLIST.

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 18. 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 160.

```
 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
 TERMTHDACT    ==> DUMP      QUIET,MSG,TRACE,DUMP,UAONLY,UADUMP,UAIMM,UATRACE


 WHEN FINISHED, PRESS ENTER.


```

Figure 160. 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 supplied in the Language Environment SCEESAMP library. Install these in your CSD using the CSD utility program, DFHCSDUP.

For information about how to use the CLER transaction, see the *OS/390 Language Environment Debugging Guide and Run-Time Messages*.

Chapter 19. 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 Guide* 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 161.

Figure 161. 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:

Figure 162. 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:

Figure 163. CMAC transaction: 2nd example screen

235

CMAC—messages and codes display

If you use methods 2 through 6 (listed above), the initial screen is bypassed and you move directly to the message information screen.

Chapter 20. 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 "Command syntax" on page 240.

Full details of the options are given under "CMSG command options" on page 240.

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:

```
CMSG R=(T001,T002)#
T=1500#
'THE TIME IS 1500 HOURS'#
S@
```

where the ENTER key.

You can enter the options, except SEND and CANCEL (one of which must be entered last), in any order after CMSG and one or more spaces. You can abbreviate every option, except CANCEL, to a single character, and you can include one or more options in a single input. If you specify two or more options, you must follow each by a comma, except the last, which must be followed by a space or by an end-of-data (EOD) condition indicating the end of your input. Any data following a space in the same input is ignored.

To mark the end of a single input:

- Press the ENTER key on display devices such as the IBM 3270
- Press carriage return for hardcopy terminals such as the IBM 2741.

To maintain the conversation with a terminal, the status of all inputs, except the final input which includes the SEND option, is saved in a single temporary-storage record.

A message-switching transaction remains conversational until ended. This means that all input (even other transaction identifiers) is passed to the message-switching program. You must end a message-switching transaction (possibly involving multiple inputs) before you can start a different transaction.

Canceling a message

You can cancel your current message-switching transaction by entering:

- CANCEL as the last 6 characters of your input. You get the response:

      TERMINATED BY OPERATOR.

- CMSG as your first 4 characters, followed by end-of-data (EOD). You get the response:

      NO INPUT - REENTER.

- CMSGx as your first 5 characters, where x is any character other than a space or a C. You get the response:

      SPACE MUST FOLLOW TRANSID.

- CMSG as your first 4 characters followed by a space and EOD. This starts a new message-switching transaction. You get the response:

      CONTINUE INPUT.

The first three examples in this list terminate the conversation and allow you to enter other transactions.

Message-switching continuation

The message-switching transaction appears to be conversational. Internally, a new task is created to process each message switching input.

If a task requires more information to complete the transaction, the current status of the transaction is saved in temporary storage, a response is sent to your terminal, and the task terminates with a requirement that the next task started be a message-switching task.

The saving of status replaces any previously saved status for any message-switching transaction from that terminal.

If a message-switching transaction completes successfully, or you have canceled it, an appropriate response is sent to your terminal, the status of the transaction is not saved, and the task terminates with no transaction restart specified. The saved status is always used by a message-switching task to continue a conversational input from your terminal. You can recall the saved status by entering the transaction identifier to which you have added C (for example CMSGC). You can include additional input, but it must follow CMSGC with no intervening spaces.

This method is useful because you can enter CANCEL to bypass errors in the current input and yet not allow the conversation to continue. You can then enter, for example CMSGC and then enter EOD, which puts the terminal in the conversation that prevailed at the time of the most recent input that has not been canceled or has not resulted in a message being routed. At this point you could also add an option if desired, for example CMSGCT=1030. (Note that

there are no spaces between the transaction identifier, the C, or the following data.)

For example, enter:

    CMSG R=(T40,T41)

and you receive:

    R OK CONTINUE INPUT,
    T=1500
    RT OK CONTINUE INPUT
    'THERE WILL BE .... AT THE
    RT OK CONTINUE MSG
    MAIN NIO;FOMH PM VS;OGPTMOS SBR, CANCEL (see Note)
    TERMINATED BY OPERATOR
    CMSGCMAIN BUILDING ON CALIFORNIA AVENUE
    RT OK CONTINUE MSG
    IN ROOM 407',SEND
    MRTS OK MESSAGE HAS BEEN ROUTED

Note: You can cancel and reenter instead of trying to correct the input. This terminates the entire transaction. To save the previously entered input, enter CMSGC to restart at the previously saved status point and continue the message.

Your terminal is free between a message-switching response and the next input, and if it is in TRANSCEIVE status, an automatically initiated task can be started on your terminal before you can continue the message-switching transaction.

Upon completion of this new task, even if it required a response from you, the message-switching transaction continues as if nothing had intervened. However, if you receive a response indicating that the intervening task has issued a RETURN, you must enter the transaction identifier followed by C to force continuation of the previous message-switching transaction.

Message delivery

You can deliver a message to a destination if the following conditions are met:

- The specified delivery time has been reached
- The terminal is INSERVICE, and either:
    - The opid of the operator signed-on matches the requested opid if specified in the ROUTE option

      or
    - The class of the operator signed-on matches any class specified by the OPCLASS option.

The processing and paging status of a terminal, which also affects delivery of eligible messages, is initially specified on the CEDA DEFINE TYPETERM command. You can change the status using the CEMT, CEST, and CEOT transactions.

When the processing status is RECEIVE or TRANSCEIVE, a message is sent automatically to the terminal when no transaction is attached to it.

When the processing status is TRANSACTION, you must request message delivery by entering a paging command, see Chapter 23, "CSPG—page retrieval" on page 259.

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 243), 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.

Command syntax
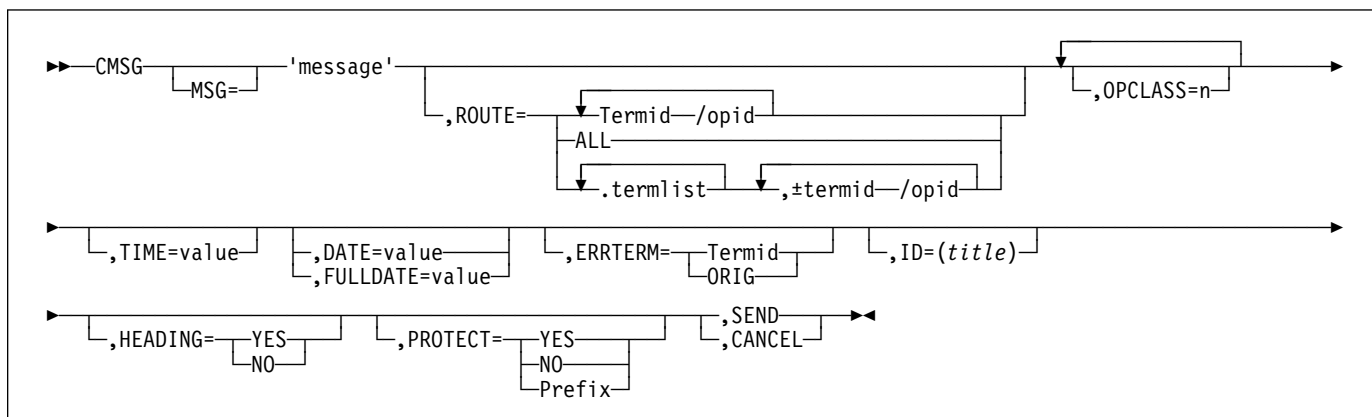
CMSG command options

The message-switching options are listed below. Except for CANCEL, you can specify the first letter of each option instead of the entire option.

CANCEL

specifies that the current input is to be ignored and institutes a **non**-conversational status between the terminal and the message-switching transaction. CANCEL must be the last 6 characters of the input. CANCEL is also effective within a message.

DATE=value

The date on which you want your message to be delivered. It can be specified in any of the following forms:

yy.ddd   year (00–99) and day (001–366).

mm/dd/yy   month (01–12), day (01–31), and year (00–99).

mm/dd   month (01–12) and day (01–31).

+**d**   number of days (0–4).

The first three of these forms provide ways of specifying absolute dates, with the year (where used) in a 2-digit format. For example, if the current system[1] date is in the year 1997, January 31 1997 could be specified as 97.031, 01/31/97, or 01/31. In this last case, the year of the current system[1] 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[1] 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 244. 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[1] 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.

---

[1] 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.

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 todays 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[1] 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[1] 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[1] 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[1] 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 244. The effects of TIME= and FULLDATE= together are as follows:

- If you specify neither a time nor a date, the message is transmitted as soon as the receiving terminal is free.

- If you specify a time but no date, the message is transmitted at the specified time today. For example, if the time now is 09.00 and you specify TIME=0930, or TIME=+30, the message is transmitted at 09.30 today.

- If you specify a date but no time, the message is transmitted at the current system[1] 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

---

[1] 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.

| 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[1] 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 23, "CSPG—page retrieval" on page 259
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 246.

Termid

is the identifier or identifiers of the terminals to which the message is to be routed. For example, ROUTE=(LA04,OL,SF2) routes the message to the three terminals with the identifiers LA04, OL, and SF2. If routing is performed to several terminals of the same device and map suffixes, CICS processes the message identically for all of them and the most restricting page size prevailing is used.

The length of the terminal identifier specified in a message-switching transaction must be in the range 1–4 characters, and must not contain any of the following characters:

/ slash

, comma

) right parenthesis

(left parenthesis

+ plus sign

– minus sign

* asterisk

blank.

Note: A single message can be delivered more than once to the same terminal. For example, the instruction ROUTE=(T001,T001) causes two transmissions of a single message to terminal T001. If the destination terminal is in TRANSCEIVE status, the message appears consecutively at the terminal. If the terminal is in TRANSACTION status, the operator must request delivery of the message.

/opid

is a 1-to 3-character operator identifier preceded by a slash. The message is routed to the first terminal at which an operator with that identifier is currently signed on. For example, ROUTE=/PJ routes the message to the first terminal found (and only the

first) with the operator identifier PJ currently signed on. If no such terminal is found, the sending operator is notified. The operator identifier that you specify must not contain any of the following characters:

, comma

) right parenthesis

space.

Termid/opid

is a terminal identifier qualified by an operator identifier to restrict the message delivery to the specified operator at the terminal location. For example, ROUTE=(LA04,OL/LBS,SF2) routes the message to terminals LA04 and SF2. The message is routed to terminal OL only if the operator whose identifier is LBS is signed on at that terminal.

ROUTE=(T001,T001/OP1,/OP1) causes the same message to be delivered three times to the same destination if the operator OP1 is signed on at T001.

ALL

causes the message to be broadcast to all terminals.

There is a variable limit on the number of terminals to which a message can be sent. This limit depends on a combination of factors. Significant factors are the types of terminal in use, the number of each type, and the length of message sent. The CMSG transaction is abended with an abend code of ABMC if the limit is exceeded.

Note: If a CMSG ROUTE=ALL is issued to a large number of terminals, a task for each terminal is initiated up to the MAXTASK value. Because the tasks are single threaded, they are suspended and can give rise to an SOS condition. For guidance about avoiding this, see the *CICS Performance Guide*.

.termlist

is a 1-or 2-character terminal list table (TLT) suffix preceded by a period. For example, .H3 identifies the terminal list table DFHTLTH3. A maximum of 10 terminal lists can be specified, and the terminal lists that you specify are merged together. The entries in the terminal lists contain terminal identifiers, or operator identifiers, or both. Duplicate entries within a single TLT are kept, though entries that are duplicated among the lists are deleted. (Entries are considered duplicate if each has the same terminal identifier and operator identifier.)

Here are two examples that show the effects of merging TLTs that contain duplicate entries. For these examples, assume that terminal list table DFHTLTL1 contains T001 twice, and that DFHTLTL2 contains T001 and T001/OP1.

- If you specify ROUTE=(.L1,.L2), all entries from DFHTLTL1 are included as destinations. Duplicate entries within DFHTLTL1 are kept. All entries from DFHTLTL2 are checked for duplicates against the entries in the previously specified DFHTLTL1 and, if a duplicate is found, it is not repeated.

  The resulting destination list is T001, T001, T001/OP1.

- The order in which you specify the TLTs is significant. If you specify R=(.L2,.L1), the DFHTLTL2 entries T001 and T001/OP1 are included in the destination list. However, the two entries for T001 in DFHTLTL1 are not included because T001 is already in DFHTLT2. In this case, the resulting destination list is T001, T001/OP1.

(±*termid/opid,...*)
A +termid/opid adds the specified destination (if not a duplicate) to the destinations contained in the requested TLT. A –termid/opid deletes the specified destination from the requested TLT. A –termid, without an opid, deletes all destinations of that terminal (with or without operator identifier) resulting from the requested TLT. + or –termid/opid parameters affect only those entries that result from requested TLTs, and have no effect on other + or – termid/opid parameters in the same request. All TLT suffixes must be entered before any + or – parameters.

Here are some examples that show the effects of specifying both TLTs that contain duplicate entries and ± entries. For these examples, assume that terminal list table DFHTLTL1 contains T001 twice, and that DFHTLTL2 contains T001 and T001/OP1.

- ROUTE=(.L1,.L2,+T001) has the same effect as R=(.L1,.L2). The entry +T001 is not added, because it is a duplicate of an entry from DFHTLTL1. The resulting destination is T001, T001, T001/OP1.

- ROUTE=(.L1,.L2,+T001/OP1,–T001) does not add +T001/OP1 because it is a duplicate of an entry in DFHTLTL2. The –T001 causes all entries from TLTs that refer to T001 (regardless of whether they are qualified by an operator identifier) to be deleted. The message ALL ROUTE ENTRIES DELETED is issued.

  If DFHTLTL2 did not contain the entry T001/OP1, the +T001/OP1 instruction would cause that entry to be added to the destination list. The –T001 instruction would not then delete the T001/OP1 entry from the list, because the effects of the + and – instructions are not cumulative: they act in isolation on the original concatenated TLTs.

- ROUTE=(.L1,.L2,–T001,+T001/OP1); the –T001 causes all entries from the TLTs that refer to T001 (including the T001/OP1 entry in DFHTLTL2) to be deleted. The +T001/OP1 entry is then added and becomes the only resulting destination. There is no duplicate because it has just been deleted.

A ROUTE option may be divided across multiple consecutive inputs. However, if it refers to a TLT, it must be completed in the same input in which it was started. An individual ROUTE parameter (termid/opid) may not be split across two inputs.

When both ROUTE and OPCLASS are specified together, OPCLASS further restricts the message transmission. For example, ROUTE=(LA04/PJL,/MGK,OL), OPCLASS=4 routes the message to terminal LA04 if the operator whose identifier is PJL is signed on. The message is also sent to the first terminal with the operator whose identifier is MGK signed on. An operator whose class is 4 must be signed on to OL before the message can be routed there. Note that the OPCLASS value is acted on only when no operator identifier is specified.

SEND
specifies that all of the options have been entered and that the message is to be routed. SEND is the final option and must be followed by a space or an end-of-data.

TIME=value
"value" is the time at which you want the message to be delivered. You can specify the time in one of the following four ways:

hhmm where "hhmm" is an absolute time in the range 0001–2400. For example, TIME=1145 causes the message to be transmitted at 11.45 am. The minutes value must be less than 60.

+**hhmm** where "hhmm" is the number of hours and minutes from the current time. The minutes value must be less than 60. For example, TIME=+0720 means that the message is to be transmitted in 7 hours and 20 minutes from now. A value of TIME=+2400 means the same as DATE=+1.

+**mm** where "mm" is the number of minutes from the current time. This value must be in the range 0–99. So, for example, a value of TIME=+75 causes the message to be transmitted 1 hour and 15 minutes from now. The values TIME=+90 and TIME=+0130 both cause the message to be transmitted in 90 minutes time.

+**m** where "m" is the number of minutes from the current time. This value must be in the range 0–9. So, for example, a value of TIME=+5 causes the message to be transmitted 5 minutes from now.

If you specify a delivery time on the current day that falls within the past hour, it is interpreted as a request for immediate delivery. An earlier time than that is considered already passed and is treated as an error. The following message is issued:

`TIME ALREADY PASSED`

Note that, if the current time is 00.15, T=2345 is interpreted as 23.45 today because there has been a change of date. The message is not therefore transmitted immediately.

Logical unit destinations

This section describes the use of the CMSG transaction to send messages to logical units. For details of message handling within CICS subsystems, and of the use of the message-switching transaction at subsystem terminals, see the appropriate CICS/OS/VS subsystem guide.

Each logical unit in a CICS-SNA network is identified by a single terminal identifier and, if the logical unit is capable of receiving message text, messages may be routed to it in the same way as they are routed to non-SNA terminals. Routing by operator identifier may also be employed if the logical unit supports operator signon, and SNA and non-SNA destinations may be specified in the same ROUTE option.

The destination for a message sent to a logical unit can be a display or printer device, or it can be a data set or an application program in a subsystem controller. To the message sender, the destination behaves like a "terminal", and any necessary formatting is performed by the CMSG transaction or within the subsystem controller itself.

Logical device codes

Some types of logical unit (LU) can be used to get access to more than one resource within a subsystem. For example, data sent to a 3601 LU may be intended for an IBM 3604 Keyboard Display, an IBM 3618 Administrative Line Printer, or some other element of the IBM 3600 Finance Communication System. The facility provided by CICS to permit destination selection within LUs of this type is the logical device code (LDC).

The LUs that support destination selection by LDC are:

- 3601 LU
- 3770 batch LU
- 3770 batch data interchange LU
- 3790 batch data interchange LU.

For the user of the message-switching transaction, the LDC is a 2-character mnemonic code whose meaning is defined by the CICS installation. It may be used to qualify an LU destination by including it in the ROUTE option in the syntax diagram that follows:

```
►►──ROUTE=──────────────────────────────────►

►──┬──Termid──*ldc──/opid──┬──────────────►◄
   ├──ALL──*ldc─────────────┤
   │                        │
   └──.termlist──*ldc──┤ Idloop ├──┘
```
Idloop:
```
├──┬──────────────────────────────┬──┤
   └──,±termid──*ldc──/opid────────┘
```
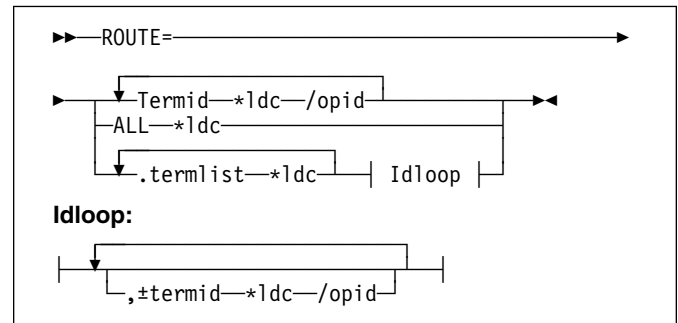
where:

*ldc

is a 2-character LDC mnemonic preceded by an asterisk (*) that qualifies the destinations. The *ldc parameter may qualify an LU identifier (termid), a general broadcast (ALL), or a terminal list table specification (.termlist). The *ldc parameter applies only to LUs, not to any start-stop or BSC terminals.

Different LDC mnemonics may be included in one ROUTE option specification; however, all destinations for one message must indicate the same device type.

Termid*ldc

associates an LDC mnemonic with an LU identification.

ALL*ldc

is a general broadcast to all terminals with the same LDC mnemonic qualifying all LUs.

Termlist*ldc
.termlist*ldc

qualifies all entries in this terminal list table with the specified LDC mnemonic. This overrides any LDC specified within the terminal list table. This LDC specification does not apply to start-stop or BSC terminals.

Note: This qualification of a TLT occurs before any succeeding TLTs or +/– entries are processed, see example 9 in "Examples of 3600 and 3770 batch destinations" on page 246.

+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 238.

    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 xxxx

A HEADING option may only be specified as follows:

```
HEADING
HEADING=YES
HEADING=NO
H
H=YES
H=NO
```

ID OPERAND GT 62 CHARS

An ID option must be from 1–62 characters in length, and enclosed within parentheses.

ID OPERAND MISSING

The ID option requires the identifier to be enclosed within parentheses.

ID PARAM MISSING OR NO)

The ID option must be from 1–62 characters in length and enclosed within parentheses.

INVALID LDC AT xxxxxxxx

The LDC mnemonic is not valid for the logical unit, or a previous entry contained an LDC mnemonic that indicated a different device type.

Note: A missing LDC mnemonic can cause this message if the default mnemonic for that station indicates a different device type than a previously indicated mnemonic (specified or defaulted).

INVALID OPND AT xxxxx

An option xxxxx is invalid (possibly caused by a keying error). No data beyond this point was processed. The status indicators represent the options that are in effect.

INVALID TBL SFX AT xxxxx

A terminal list table suffix must be 1 or 2 characters preceded by a period.

INVLD OPERATR ID AT xxxxx

An operator identifier must be 1, 2, or 3 characters preceded by a slash.

INVLD SEPARATOR AT xxxxx

The end of an option must be followed by a space, a comma (except for SEND), a carriage return (new-line character), or end of data (EOD).

INVLD TERMINL ID xxxx

There is no entry in the CICS terminal definition for the terminal identifier specified for a ROUTE destination. The valid operator reply is one of the following:

1. A terminal identifier, or operator identifier, or both, in the same format as a normal termid/opid ROUTE

destination (without entering the option ROUTE= or R=) that replaces the invalid destination.

2. A space as the first character, which causes the invalid destination to be deleted.

3. Any option that causes a return to normal option processing mode. If another ROUTE option is not entered, the same condition exists after the SEND option is reentered.

Note: 1 and 2 cause the message to be routed if the resulting destinations are valid; otherwise, another error message is issued.

LDC NOT 2 CHARS AT xxxxx

The LDC mnemonic that follows the * in the ROUTE destination must be 2 characters.

MESSAGE HAS BEEN ROUTED

A SEND option has been processed and the message has been routed to its destinations. This is the normal completion of a message-switching transaction.

MSG NOT SENT - NO RECEIVER

This condition is caused by an OPCLASS option specified without a ROUTE option and no operators, with the specified operator class, or classes, currently signed-on to terminal types supported by BMS. To correct this situation, a new destination must be entered by the ROUTE option, or OPCLASS option, or both.

NEED OPCLASS OR ROUTE

The SEND option was entered without specifying any destinations for the message. To correct this situation, a destination must be entered by the ROUTE option, or OPCLASS option, or both.

NO INPUT - REENTER

A transaction identifier was entered followed by end-of-data (EOD), or the number of backspace characters entered caused the deletion of all the data (including the position following the transaction identification, if present). If the input does not start with the transaction identifier, the terminal remains in conversation. Otherwise, it is not in conversation.

NO MESSAGE PROVIDED

The SEND option was entered without specifying the message. To correct the situation, provide the message.

NO MSG STRT QUOT AT xxxxx

A MSG parameter must be enclosed within single quotation marks. A single quotation mark must follow an M or MESSAGE.

NO TRMID FOR LDC AT xxxxx

The destination started with an asterisk *. A logical unit identification must precede the LDC mnemonic.

NO TERM LIST TBL-xx

An entry for a terminal list table with suffix xx (DFHTLTxx) was not found in the CICS system definition.

OPCLASS NOT 1-24 AT xxxxx

An OPCLASS parameter must be in the range 1–24.

OPERATORS NOT SIGNDON nnn

The ROUTE option specified operator identifications without terminal identifiers (that is, ROUTE=/opid). nnn operator identifications are not currently signed on.

The valid reply to this is to enter one of the following:

1. A space as the first character, which deletes the nnn destinations.

2. The character D (or any other character) which causes each of the nnn destinations to be individually displayed through the message OPRTR ID NOT SIGNDON.

3. Any option that causes a return to normal option processing mode. If no further ROUTE options are entered, this same condition exists if an operator does not sign on before the SEND option is reentered.

Note: This message is only issued if nnn is greater than 5; otherwise, each destination is individually displayed. Reply 1 causes the message to be routed if the resulting destinations are valid; otherwise, another error message is issued.

OPRTR ID NOT SIGNDON /xxx

An operator with identifier xxx, specified as a destination (entered either by the terminal operator or included as an entry in a TLT whose suffix was specified), is not currently signed on to any terminal controlled by CICS.

The valid reply to this is to enter one of the following:

• A terminal identifier, or operator identifier, or both in the same format as a normal termid/opid ROUTE destination (without entering the option ROUTE or R), which replaces this /xxx destination.

• A space as the first character, which causes the /xxx destination to be deleted.

• Any option that causes a return to normal option processing mode. If no further ROUTE options are entered, this same condition exists if an operator with identifier xxx does not sign on by the time the SEND option is reentered.

Note: The first two replies cause the message to be routed if the resulting destinations are valid, otherwise, another error message is issued.

PROTECT OPND NOT 1-2 CHAR

The PREFIX option of the PROTECT option must not specify more than 2 characters.

ROUTE INVALID AT xxxxx

A field separator entered before the first ROUTE parameter, or 2 consecutive field separators were found. Reenter a valid ROUTE option.

SPACE MUST FOLLOW TRANSID

One or more spaces must follow the transaction identifier before the first option can start. (The character C causes a continuation from the last input preceding a normal completion or a cancelation.)

TERMID GT 4 CHAR AT xxxxx

A terminal identifier must be 1–4 characters.

TERMID NOT +/– at xxxxx

All TLT suffixes must be specified before any individual destination (which must be preceded by a + or –) may be entered. A ROUTE option that specifies a TLT suffix may not be continued on a succeeding input. If a ROUTE option includes more than one parameter, it must be enclosed in parentheses.

TERMINATED BY OPERATOR

The operator entered CANCEL as the last 6 characters of the previous input, or pressed the CLEAR key twice in succession on an IBM 3275 (or 3277) display station. The previous input is canceled and the terminal is **not** in conversational mode with CICS message switching.

TIME ALREADY PASSED

A delivery time for the current day, more than one hour before the current time is considered already passed and is invalid.

If the message is for immediate delivery, enter T=+0, to replace the TIME option that is still in effect.

If the message is for delivery at the specified time on a future day, enter the appropriate DATE option. The specified time is still in effect.

If the message is for delivery at a different time, reenter the TIME option.

TIME INVALID AT xxxxx

TIME must be one of the following:

hhmm 0001–2400 (mm<60)

+**hhmm** 0000–2400 (mm<60)

+**mm** 00–99

+**m** 0–9

where hh=hours and mm=minutes.

TIME NOT 4 CHARS AT xxxxx

A TIME parameter must be a 4-digit number if it is not preceded by +.

TOO MANY TBL SFX AT xxxxx

A maximum of 10 TLT suffixes may be specified.

UNBALANCD PARENS AT xxxxx

A ROUTE or OPCLASS option started with a parenthesis but did not have one at the end, or ended with a parenthesis but did not have one at the beginning. xxxxx represents the characters near the relevant end of the option.

UNSUP OPRT TRMNL xxxxxxxx

An operator identifier without a terminal identifier was specified as a ROUTE destination; however that operator is signed on to a terminal that is a type not supported by BMS and is, therefore, an invalid destination.

The valid reply to this is the same as for the message OPRTR ID NOT SIGNDON.

UNSUP TERMINAL xxxxxxxx

A terminal identifier, specified as a ROUTE destination, is for a terminal type not supported by BMS and is, therefore, an invalid destination.

The valid reply to this is the same as for the message INVLD TERMINL ID.

ZERO LENGTH MSG AT ' '

A message must have at least one character enclosed in single quotation marks. A single quotation mark to be included as part of a message must be entered as two separate single quotation mark characters. A message must start with a single quotation mark and may be followed immediately by pairs of single quotation marks.

+**DATE INVLD 2330 TO 0030**

A DATE=+ option is invalid from 30 minutes before to 30 minutes after midnight.

Examples of complete message-switching transactions

Here are some typical examples of the CMSG transaction.

In all of the examples,

1. CMSG R=ALL,'GOOD MORNING#TODAY IS FRIDAY FEB 1',S

This sends a good morning message to all terminals.

2. CMSG 'GOOD MORNING',R=.G1,T=0900,S

This routes a good morning message to be delivered at 0900 hours to all destinations contained in DFHTLTG1. (Locations in Eastern time zone.)

3. CMSG 'GOOD MORNING',R=(.G2,+CHI4),T=1000,S

This message is for delivery at 10.00 hours to terminals in Central time zone (DFHTLTG2). Also included as a destination is termid CHI4, which was recently added but not yet included in the TLT.

4. CMSG 'SUPERVISORS' 'MEETING IN 5 MINUTES',O=(4,7),S

This routes the message to all supervisors who are assigned operator class 4 or 7 and are currently signed on to CICS.

5. CMSG R=ALL,'NEW PRICE MODEL 402 - $28.70',S

This sends the price of the new model to all terminals.

6. CMSG 'FRED - WHAT IS CURRENT STOCK ON P/N 4837-LP4# BPK',R=BLD3,H,S

This routes a message to BLD3 asking Fred a question and including the requestor's initials (operator identifier). A heading is included so that Fred also knows the originating termid.

7. CMSG 'PNO 4837-LP4, QTY 26?',R=/BPK,H,S

This shows Fred's answer routed to the terminal at which operator identifier BPK is currently signed on.

8. CMSG 'BRID KADDER ARRIVING 3:15 PST. PLEASE MAKE RESERVATION# M. KRN',R=PADC,H,S

This makes a request for a reservation for a Mr Kadder.

9. CMSG 'SHUTDOWN IN 5 MINUTES',R=ALL,T=1655,S

This sends the shutdown message to all terminals.

Note: Examples 1, 2, 3, and 9 are possible candidates for inclusion in a sequential (BSAM) terminal input stream during startup.

Chapter 21. 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:

```
►►──CRTE──SYSID=sysid─────────────────►◄
                     └─,TRPROF=trprof─┘
```

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.

CRTE—remote transactions

Chapter 22. 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*.

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.

System spooling interface trace

You can use CSFE to request additional trace and diagnostic facilities for the system spooling interface.

```
►►—CSFE spoolfe=——ON——►◄
                 └—OFF—┘
```
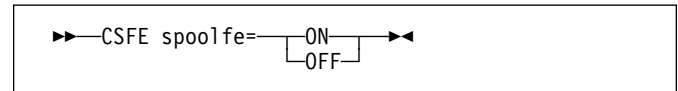
OFF
switches off the system spooling interface trace and the diagnostic facility.

ON
specifies that additional trace entries of system type are provided for the system spooling interface.

SPOOLFE
is the system spooling interface trace and diagnostic facility.

Terminal builder trace

You can use the CSFE transaction to activate and deactivate the terminal builder trace.

```
►►—CSFE zcqtrace——=termid————————►◄
                 ├—,AUTOINSTALL—┤
                 └—,OFF—————————┘
```
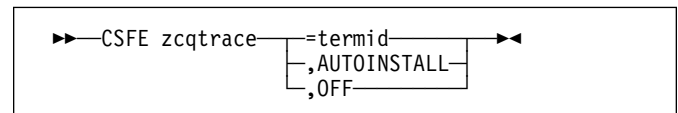
,AUTOINSTALL
Switches tracing on for autoinstalled terminals.

,OFF
Switches tracing off for individual and autoinstalled terminals.

=termid
Switches tracing on for the terminal or connection identified by termid.

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=xxxx──┬──────────────►
                └─,TASKREQ=yyy──┘

►──────────────────────────►◄
    │         ┌─OFF─┐     │
    └─,STGFRZ=─┴─ON──┘
```
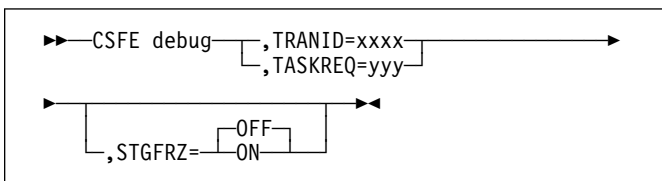
You must code the options in the order shown in the syntax display above.

STGFRZ= ON or off
> 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─┬──►◄
                │           └─NONE────┘  │
                └─,CHKSTRM=─┬─CURRENT─┐
                            └─NONE────┘
```
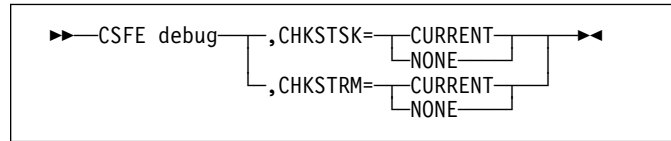
CURRENT
> Specifies that validity checks are to be performed on the user storage (not the CICS storage) for the current task whenever the trace program is entered. Validity checks are to be performed for the currently active task, as follows:
>
> • Validity checking of the storage chain for the task
>
> • Validity checking of the addresses and storage accounting areas for all storage areas on the transaction storage chain.
>
> 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.

```
►►──CSFE DEBUG──┬─,TRAP=──┬─ON──┬──────►◄
                │         └─OFF─┘
                └─,GTRAP=──┬─ON──┐
                           └─OFF─┘
```
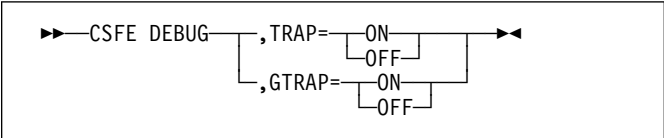
TRAP={ON|OFF}
> The global trap/trace program exit routine is to be activated or deactivated.

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.

CSFE—terminal and system test

Chapter 23. 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 232).
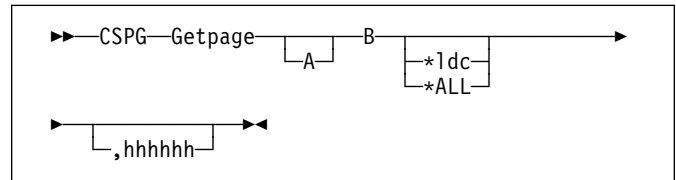
Getpage
:   This is the user-defined identification code for the retrieval transaction, and is 1–7 characters long. This code is defined by the PGRET system initialization parameter. For convenience, make the code as short as possible; for example, P/.

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" on page 261.

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 "Status of hardcopy terminal" on page 263.

    X   enter data that is not recognized as a paging command.

    This is discussed further under "Single keystroke retrieval (SKR)" on page 260.

*ldc

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 *ldc.

Hhhhhh

Is the 6-digit hexadecimal identifier of an undelivered logical message to be retrieved. You would normally determine this identifier by entering a page query command (see "Display information about logical messages" on page 261).

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 BTAM or VTAM, you can use PA or PF keys, as defined in the SKRxxxx system initialization parameter, to enter page retrieval commands, and greatly reduce the number of keystrokes necessary in a page retrieval session.

After the session has ended, the PA and PF keys can again be used for initiating transactions in the usual way. The *CICS System Definition Guide* gives further information on this topic.

Page retrieval for logical devices

Your application program can send messages to a communication system such as the IBM 3600. Your application program can also send different pages to each of the logical devices within the system. These pages must be grouped as a single logical message, held in storage.

Each part of a logical message is accompanied by a logical device code (LDC) that identifies its destination. When retrieving the pages for a logical device, you must specify the LDC for that device. However, you do not need to do this if the user-written program in the logical device controller provides a default LDC mnemonic.

If you specify the page number and *ldc, delivery begins with the specified page for that logical device. If the logical device is in AUTOPAGE status, all pages for that logical device, starting with the requested page, are written. If the logical device is in PAGE status, only the specified page is written.

If you omit *ldc, and the application program in the logical device does not provide a default, BMS selects the first page destined for a logical device that is currently in PAGE status. The application program in the logical unit should provide a default LDC, because pages may be delivered to logical devices other than those you originally specified.

In either case, if your request is the first request for a page from a logical message, all pages (within the logical message) destined for logical devices currently in AUTOPAGE status are delivered after the first or specified page is delivered to a device in PAGE status.

The getpageA command supports an *ldc specification for nonrouted messages created for a 3601 logical unit. The various forms of getpageA are interpreted as follows:

- getpageA*ALL-resets to autopaging and delivers all pages for the logical devices that have autopage specified in the terminal definition, but that were temporarily placed in page status by the CICS application programmer.

- getpageA*ldc-resets the specified logical device to autopage and delivers the pages for that *ldc. The command is similar to getpageA*ALL, except that it operates only for the specified logical device.

- getpageA-resets to autopaging and delivers the remaining pages for the logical device for which pages are currently being displayed.

Page retrieval and partitions

The page retrieval commands, including any PA and PF keys used in single keystroke retrieval, refer to the partition in which they are entered. For example, P/N means 'retrieve the next page for this partition'. The partition in which the last paging command is entered is the *current partition*. A paging command for a partitioned message entered on a nonpartitioned screen (that is, after CLEAR) refers to the current partition.

After a transition from base (unpartitioned) state to partitioned state, or a change of partition set (that is, the initial page display, a page display following CLEAR, or following a change in the level of page chaining), the *current page* is displayed in each partition. The current page is initially the first page for that partition. It is updated by page retrieval commands.

Display information about logical messages

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 259) to retrieve pages for this display.

Message chaining

Usually, when you have viewed the output pages from a transaction, there is no need to retain the pages. However, you might want to initiate a new transaction and still retain the old pages, so that you can view them again after you have displayed output from the new transaction. This is called *message chaining* and can be done using the following command:

```
►►──CSPG──Messchnyyyy──────────────►◄
                    └─Parameters─┘
```
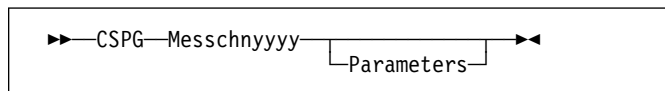
Messchn
> The installation-defined message-chaining transaction identification code (for example, C/), as defined in the PGCHAIN system initialization parameter.

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 "Message termination" on page 262. 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 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.

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:
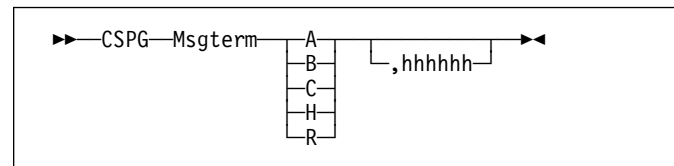
Msgterm
A user-defined transaction identifier (1–7 characters) for the message termination transaction (for example T/) as defined in the PGPURGE system initialization parameter.

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 (hhhhhh) is not specified in the termination request and there is no message currently displayed, the oldest message is automatically retrieved.

Copying pages

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.

Status of hardcopy terminal

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.

Routed pages displayed at terminal

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 20, "CMSG—message switching" on page 237. 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 24. 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:

```
►►──CWTO──Message──►◄
```

 Press the ENTER key. If the transaction runs successfully, your message is sent to 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
  ```

CWTO—write to console operator

Chapter 25. 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.*

DSNC—write to console operator

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
CAFB	CAFB1	DFHAUGRP	Transaction Affinities Utility data saver
CAFF	CAUCAFF1	DFHAUGRP	Transaction Affinities Utility controller
CATA	DFHZATA	DFHSPI	Autoinstall terminal
CATD	DFHZATD	DFHSPI	Delete autoinstalled terminal
CATR	DFHZATR	DFHSPI	Restart autoinstalled terminal
CBAM*	DFHECBAM	DFHOPER	BTS browser
CCIN	DFHZCN1	DFHCLNT	CICS Client
CDBC*	DFHDBME	DFHDBCTL	DBCTL connect/disconnect
CDBD	DFHDBDI	DFHDBCTL	DBCTL disable
CDBF	DFHD2CM3	DFHDB2	CICS DB2 attachment facility 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	Dynamic routing of a delayed non-terminal-related START request
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
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
CEMT*	DFHEMTP	DFHOPER	Master terminal
CEOT*	DFHEOTP	DFHOPER	Terminal status
CESC*	DFHCESC	DFHSIGN	Terminal timeout
CESD	DFHCESD	DFHSDAP	Shutdown assist
CESF*	DFHSFP	DFHSIGN	Sign off
CESN*	DFHSNP	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
CFSL	DFHDTLX	-	Shared DT load
CFTS	DFHZATS	DFHSPI	Remote mass flag
CGRP	-	-	Persistent sessions (See note)
CHLP	DFHCMAC	DFHCMAC	Alias for CMAC

269

Appendix

Tranid	Program	Group	Description
CIND*	DFHINDT	DFHINDT	CICS in-doubt testing tool
CIOD	DFHIIOPA	DFHIIOP	Default CICS IIOP interface started by CIOR
CIOF	DFHIIOPA	DFHIIOP	CICS Generic factory started by CIOR
CIOR	DFHIIOP	DFHIIOP	CICS IIOP interface started by SO_Domain
CITS	DFHZATS	DFHSPI	Remote install
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
CMPX	DFHMXP	DFHISC	Local queuing shipper
CMSG*	DFHMSP	DFHMSWIT	Message switching
CMTS	DFHZATS	DFHSPI	Remote mass delete
COVR	DFHZCOVR	-	CICS open VTAM retry
CPLT	-	-	PLT programs
CPMI	DFHMIRS	DFHISC	Synchronization level 1 mirror
CQPI	DFHCLS5	DFHISC	Connection Queisce Architected transaction (inbound)
CQPO	DFHCLS5	DFHISC	Connection Queisce Architected transaction (outbound)
CQRY	DFHQRY	DFHSTAND	ATI query support
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
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	DFHFCRD	-	RLS cleanup
CSFU	DFHFCU	DFHOPCLS	File open utility
CSGM	DFHGMM	DFHVTAM	"Good-morning" signon
CSHA	-	-	Scheduler services
CSHQ	-	-	Scheduler services
CSHR	DFHMIRS	DFHISC	SCHEDULER SERVICES REMOTE ROUTING
CSMI	DFHMIRS	DFHISC	Mirror transaction
CSM1	DFHMIRS	DFHISC	SYSMSG 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
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

Tranid	Program	Group	Description
CSSF	DFHRTC	DFHISC	CRTE cancel remote transaction
CSSY	—	—	(See note)
CSTE	DFHTACP	DFHSTAND	Terminal abnormal
CSTP	—	—	(See note)
CSXM	—	—	(See note)
CSZI	DFHSZRMP	DFHFEPI	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
CWBA	DFHWBA	DFHWEB	CICS Web Interface Alias Transaction
CWBC	DFHWBC00	DFHWEB	CICS Web Interface Connection Manager Transaction
CWBG	DFHWBGB	DFHWEB	CICS Web Interface Garbage Collection Transaction
CWBM	DFHWBM	DFHWEB	CICS Web Interface Server Controller Transaction
CWTO*	DFHCWTO	DFHCONS	Write to console operator
CWXN	DFHWBXN	DFHWEB	CICS Web Attach Transaction
CXCU	DFHCXCU	DFHSTAND	XRF catchup transaction
CXRE	DFHZXRE	DFHSTAND	Autoconnect processing
CXRT	DFHCRT	DFHISC	Transaction routing relay
DSNC*	DFHD2CM1	DFHDB2	CICS DB2 attachment facility transaction

Note: Transactions CGRP, CSSY, CSTP, and CSXM are transaction names that are used by some CICS tasks.

Appendix

Index

Special Characters

- and CEMT   47
 :i1.minus sign following CEMT   47
? and CEMT   47
(VALUE)
   CEMT INQUIRE JMODEL   93
   CEMT INQUIRE JOURNALNAME   94
   CEMT SET PROCESSTYPE   194
* and CEMT   49
*ALL
   CSPG   260
*LDC
   CMSG   245
–TERMID*LDC/OPID
   CMSG   245
+ in CEMT syntax   49
+ in scrolling under CEMT   50
+HHMM
   CMSG   250
+M
   CMSG   250
+MM
   CMSG   250
+TERMID*LDC/OPID
   CMSG   245

A

abbreviation of keywords
   CEMT transaction   48
abbreviations for keywords   3
ABEND
   CEMT INQUIRE DB2CONN   66, 68
   CEMT INQUIRE DB2ENTRY   70
   CEMT SET DB2CONN   172
   CEMT SET DB2ENTRY   175
access method control block (ACB)   155, 214
ACCESSMETHOD
   CEMT INQUIRE CONNECTION   61
   CEMT INQUIRE DSNAME   77
   CEMT INQUIRE FILE   88
ACCOUNTREC
   CEMT INQUIRE DB2CONN   65
   CEMT INQUIRE DB2ENTRY   69
   CEMT SET DB2CONN   170
   CEMT SET DB2ENTRY   174
ACQ
   CEOT   217
ACQSTATUS
   CEMT INQUIRE MODENAME   97
ACQUIRED
   CEMT INQUIRE CONNECTION   61

ACQUIRED *(continued)*
   CEMT INQUIRE MODENAME   97
   CEMT INQUIRE NETNAME   103
   CEMT INQUIRE TERMINAL   134
   CEMT SET CONNECTION   166
   CEMT SET MODENAME   190
   CEMT SET NETNAME   193
   CEMT SET TERMINAL   209
acquiring VTAM terminals   193, 209
ACTION
   CEMT INQUIRE DSNAME   77
   CEMT INQUIRE SYDUMPCODE   117
   CEMT INQUIRE TRDUMPCODE   138
   CEMT SET TSQUEUE   212
ACTIVE
   CEMT INQUIRE ENQ   83
   CEMT INQUIRE MODENAME   97
   CEMT INQUIRE SYSTEM   120, 121
   CEMT INQUIRE TCLASS   125
   CEMT INQUIRE UOW   146
   CEMT INQUIRE UOWENQ   152
ACTIVITY
   CEMT INQUIRE TASK   122
   CEMT SET PROCESSTYPE   194
ACTIVITYID
   CEMT INQUIRE TASK   122
ACTOPENTCBS
   CEMT INQUIRE SYSTEM   118
ADD
   CEMT INQUIRE SYDUMPCODE   117
   CEMT INQUIRE TRDUMPCODE   138
   CEMT SET SYDUMPCODE   199
   CEMT SET TRDUMPCODE   211
ADDABLE
   CEMT INQUIRE FILE   88
   CEMT SET FILE   183
ADDREXECENQ
   CEMT INQUIRE ENQ   83
   CEMT INQUIRE UOWENQ   152
ADDSTATUS
   CEMT INQUIRE FILE   88
AFFINITY
   CEMT INQUIRE CONNECTION   61
AFTER
   CEDA   38
AGE
   CEMT INQUIRE UOW   145
AGING
   CEMT INQUIRE SYSTEM   118
   CEMT SET SYSTEM   201
AKP
   CEMT INQUIRE SYSTEM   118
   CEMT SET SYSTEM   201

I

ID
 CMSG   242
ID OPERAND GT 62 CHARS
 CMSG   249
ID OPERAND MISSING
 CMSG   249
ID PARAM MISSING OR NO)
 CMSG   249
IDENTIFIER
 CEMT INQUIRE TASK   123
identifiers
 resource, lists of   49
identifying terminals   4
IDLE
 CEMT INQUIRE DELETSHIPPED   73
 CEMT SET DELETSHIPPED   177
IMMCLOSE
 CEMT INQUIRE IRC   93
 CEMT INQUIRE VTAM   155
 CEMT SET IRC   187
 CEMT SET VTAM   215
IMMEDIATE
 CEBT PERFORM SHUTDOWN   27
 CEMT PERFORM SHUTDOWN   160
IMMQUIESCED
 CEMT INQUIRE DSNAME   80
 CEMT SET DSNAME   180
INACTIVE
 CEMT INQUIRE SYSTEM   120, 121
INDEXRECFULL
 CEMT INQUIRE UOWDSNFAIL   149
INDIRECT
 CEMT INQUIRE CONNECTION   61
 CEMT INQUIRE TDQUEUE   130
INDOUBT
 CEMT INQUIRE TASK   123
 CEMT INQUIRE TDQUEUE   129
 CEMT INQUIRE TRANSACTION   136
 CEMT INQUIRE UOW   146
 CEMT INQUIRE UOWDSNFAIL   149
INDOUBTMINS
 CEMT INQUIRE TASK   123
 CEMT INQUIRE TRANSACTION   136
INDOUBTWAIT
 CEMT INQUIRE TASK   123
 CEMT INQUIRE TDQUEUE   130
 CEMT INQUIRE TRANSACTION   136
INFLIGHT
 CEMT INQUIRE UOW   146
Information Display System   239
INPUT
 CEMT INQUIRE TDQUEUE   129
 CMSG   248

inquire about a resource   47
INQUIRE keyword, CEMT   47
INS
 CEOT   217
INSERVICE
 CEMT INQUIRE CONNECTION   64
 CEMT INQUIRE LINE   96
 CEMT INQUIRE NETNAME   102
 CEMT INQUIRE TERMINAL   133
 CEMT SET CONNECTION   167
 CEMT SET LINE   189
 CEMT SET NETNAME   193
 CEMT SET TERMINAL   209
interface to master terminal, programmable   51
INTERNAL
 CEMT INQUIRE TASK   124
interregion communication (IRC)   92, 187
INTERVAL
 CEMT INQUIRE DELETSHIPPED   73
 CEMT INQUIRE STATISTICS   113
 CEMT SET DELETSHIPPED   177
 CEMT SET STATISTICS   198
INTRA
 CEMT INQUIRE TDQUEUE   130
INTTRACE
 CEBT INQUIRE INTTRACE   25
 CEBT INQUIRE transaction   25
 CEBT SET transaction   31
 CEMT SET transaction   187
INVALID
 CEMT INQUIRE DSNAME   78
INVALID LDC AT XXXXXXXX
 CMSG   249
INVALID OPND AT XXXXX
 CMSG   249
INVALID TBL SFX AT XXXXX
 CMSG   249
INVLD OPERATR ID AT XXXXX
 CMSG   249
INVLD SEPARATOR AT XXXXX
 CMSG   249
INVLD TERMINL ID XXXX
 CMSG   249
invoking a transaction   2
invoking CEMT from an application program   51
IOERROR
 CEMT INQUIRE UOWDSNFAIL   149
IOTYPE
 CEMT INQUIRE TDQUEUE   129
IRBATCH
 CEMT INQUIRE transaction   92
IRC
 CEMT INQUIRE CONNECTION   61
 CEMT INQUIRE IRC   92
 CEMT INQUIRE transaction   92
 CEMT INQUIRE UOWLINK   153

Sending your comments to IBM

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book.

Please limit your comments to the information in this book and the way in which the information is presented.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

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

You can send your comments to IBM in any of the following ways:

* By mail, to this address:

   Information Development Department (MP095)
   IBM United Kingdom Laboratories
   Hursley Park
   WINCHESTER,
   Hampshire
   SO21 2JN
   United Kingdom

* By fax:

   – From outside the U.K., after your international access code use 44–1962–870229

   – From within the U.K., use 01962–870229

* Electronically, use the appropriate network ID:

   – IBM Mail Exchange: GBIBM2Q9 at IBMMAIL

   – IBMLink™: HURSLEY(IDRCF)

   – Internet: idrcf@hursley.ibm.com

Whichever you use, ensure that you include:

* The publication number and title

* The topic to which your comment applies

* Your name and address/telephone number/fax number/network ID.

IBM ®

Program Number: 5655-147

SC33-1686-35

IBM

CICS TS for OS/390

CICS Supplied Transactions

Release 3