

Cincom

SUPRA SERVER PDM

CICS Connector Systems Programming
Guide (OS/390 & VSE)

P26-7452-64



SUPRA[®] Server PDM CICS Connector Systems Programming Guide (OS/390 & VSE)

Publication Number P26-7452-64

© 1992–1994, 1997, 1998, 2000, 2002 Cincom Systems, Inc.
All rights reserved

This document contains unpublished, confidential, and proprietary information of Cincom. No disclosure or use of any portion of the contents of these materials may be made without the express written consent of Cincom.

The following are trademarks, registered trademarks, or service marks of Cincom Systems, Inc.:

AD/Advantage [®]	iD CinDoc [™]	MANTIS [®]
C+A-RE [™]	iD CinDoc Web [™]	Socrates [®]
CINCOM [®]	iD Consulting [™]	Socrates [®] XML
Cincom Encompass [®]	iD Correspondence [™]	SPECTRA [™]
Cincom Smalltalk [™]	iD Correspondence Express [™]	SUPRA [®]
Cincom SupportWeb [®]	iD Environment [™]	SUPRA [®] Server
CINCOM SYSTEMS [®]	iD Solutions [™]	Visual Smalltalk [®]
	Intelligent Document Solutions [™]	VisualWorks [®]
gOOj [™]	Intermax [™]	

UniSQL[™] is a trademark of UniSQL, Inc.
ObjectStudio[®] is a registered trademark of CinMark Systems, Inc.

All other trademarks are trademarks or registered trademarks of their respective companies.

Cincom Systems, Inc.
55 Merchant Street
Cincinnati, Ohio 45246-3732
U.S.A.

PHONE: (513) 612-2300
FAX: (513) 612-2000
WORLD WIDE WEB: <http://www.cincom.com>

Attention:

Some Cincom products, programs, or services referred to in this publication may not be available in all countries in which Cincom does business. Additionally, some Cincom products, programs, or services may not be available for all operating systems or all product releases. Contact your Cincom representative to be certain the items are available to you.

Release information for this manual

The *SUPRA Server PDM CICS Connector Systems Programming Guide (OS/390 & VSE)*, P26-7452-64, is dated January 15, 2002. This document supports Release 2.7 of SUPRA[®] Server PDM in IBM mainframe environments.

We welcome your comments

We encourage critiques concerning the technical content and organization of this manual. Please take the [survey](#) provided with the online documentation at your convenience.

Cincom Technical Support for SUPRA Server PDM

FAX: (513) 612-2000
Attn: SUPRA Server PDM Support

E-mail: helpna@cincom.com

Phone: 1-800-727-3525

Mail: Cincom Systems, Inc.
Attn: SUPRA Server Support
55 Merchant Street
Cincinnati, OH 45246-3732
U.S.A.



Contents

About this book	ix
Using this document.....	ix
Document organization	x
Revisions to this manual	xi
Conventions	xii
SUPRA Server documentation series	xiv
Understanding the CICS Connector	17
How the CICS Connector works	18
CICS tasks and the CICS Connector	19
CICS transactions and the CICS Connector	21
Connecting and disconnecting the PDM	23
How the CICS Connector allocates storage.....	26
How the CICS Connector handles task recovery.....	28
How to retry PDML	31
How operator commands are processed	32
How SUPRA Server components and related products work with CICS and the CICS Connector.....	34
The Relational Data Manager (RDM) and CICS	35
SPECTRA and CICS.....	37
MANTIS and CICS	39
Directory maintenance and CICS.....	40
How PDM operating modes work with CICS.....	41
Central operating mode.....	41
Attached operating mode	43
Attached central operating mode	44
Operating the CICS Connector	45
Connecting the CICS Connector to the PDM.....	46
Disconnecting the CICS Connector from the PDM	47
Determining the status of the CICS Connector	49
Opening and closing database files dynamically.....	50
Purging a task	52

Resolving problems	53
Common SUPRA Server and CICS error messages and symptoms.....	54
CICS environment setup problems.....	55
Debugging tools.....	56
CICS Connector Activity Audit Trail	56
CICS trace table entries.....	58
ABEND operator control command	59
DUMP operator control command.....	59
System recovery	60
CICS abends	60
PDM abends	60
Concurrent CICS and PDM abends.....	60
Installing the CICS Connector	61
Defining connector resources to CICS	62
Transient data destination	63
VSAM files	66
Programs and Transactions	67
Program List Table (PLT)	68
System Initialization Table (SIT)	70
Transaction List Table (XLT)	70
Modifying your CICS execution JCL	71
Generating the operator control parameter table (CSTXOTBL).....	72
Tailoring RDM options	74
CICS Steplib requirements	75
Tailoring the CICS Connector node error program.....	76
Tailoring NEP for SUPRA.....	76
Example MRO environment	79
Using operator control commands	83
ABEND command	84
CLOSE command.....	85
CONNECT command.....	87
DISCONNECT command.....	91
DUMP command	93
OPEN command.....	94
PURGE command	96
RDM command.....	97
STATUS command.....	98

Using Cincom CICS Connector exits	99
Overview of the user exits	100
CICS Connector macros	102
The TUID user exit	104
The CSTXUDAT user exit	107
The TPRP and TPOP user exits	110
The TPRP (preprocess) user exit	110
The TPOP (post-process) user exit	113
The TPRE and TPOS user exits	116
The TPRE (pre-DML) exit	116
The TPOS (post-DML) user exit	118
The TSYN (Task Synchronization) user exit	122
The TDUE (Task Detach) user exit	124
The CSTXMRXT exit	127
Obsolete user exits	128
Using Cincom-supplied transactions	129
Overview of supplied transactions	129
ISRA PDM information menu	129
ISRC PDM File Information	130
ISRD PDM File usage Information	130
ISRF PDM Connector Information	131
ISRB PDM Information	132
ISRE Secondary Key Information	132
Cincom-supplied source programs	133
CSTXGNIT	133
CSTXLST1	133
CSTXUSRX	133
DFHZNEP	133
Using Cincom CICS Connector CSTXOPRM macro	135
Assembling the CSTXOPRM macro	135
Format 1 (define default CONNECT values)	136
Format 2 (group files)	151
Format 3 (define files)	152
Using Cincom CICS Connector C\$TXUNCP macro	155
Assembling the C\$TXUNCP macro	155
Index	157

About this book

Using this document

This manual is for systems programmers and database administration personnel with detailed knowledge of the Customer Information Control System (CICS). This manual provides instructional and reference material about the operation of SUPRA[®]. The CICS Connector between CICS and the SUPRA Physical Data Manager (PDM) runs with a central, attached, or attached central PDM and supports OS/390, and VSE/ESA operating systems.

Document organization

The information in this manual is organized as follows:

Chapter 1—Understanding the CICS Connector

Provides an introduction to what the CICS Connector is and how it works.

Chapter 2—Operating the CICS Connector

Describes how to use the CICS Connector to make it possible for CICS application programs, SUPRA Server components such as Directory Maintenance and RDM, and other applications such as SPECTRA™ and MANTIS® to communicate with the PDM.

Chapter 3—Resolving problems

Describes how to use debugging tools to identify problems, their locations, and their resolutions.

Chapter 4—Installing the CICS Connector

Describes how to install the CICS Connector.

Chapter 5—Using operator control commands

Describes how to issue operator control commands.

Chapter 6—Using Cincom CICS Connector exits

Describes user exits provided for the CICS Connector.

Chapter 7—Using Cincom-supplied transactions

Describes transactions provided with the CICS Connector.

Chapter 8—Cincom-supplied source programs

A description of the programs supplied in source.

Appendix A—Using Cincom CICS Connector CSTXOPRM macro

Describes how to use the CSTXOPRM macro.

Appendix B—Using Cincom CICS Connector C\$TXUNCP macro

Describes how to use the C\$TXUNCP macro.

Index

Revisions to this manual

The following summarizes the changes to this manual:

- ◆ This release is for CICS ESA only. All information about pre-ESA releases has been removed.
- ◆ The Cincom Task Related User Exit (TRUE) has been changed so that tasks that do not call SUPRA (datbas) will not drive Cincom's TRUE. In previous releases all tasks in CICS drove our TRUE.
- ◆ Inflight connections to SUPRA that need to be cleaned up because of programming problems or task abends no longer start OPER as a nonterminal task for cleanup. The cleanup is done as part of the normal task termination.
- ◆ Connector memory is acquired from CICS DSA for all versions (OS/390 or VSE).
- ◆ The description of the CSTXOPRM macro parameters has been moved from the *PDM and Directory Administration Guide*, P26-2250, to this manual. The following changes have been made to the CSTXOPRM macro:
 - The IMMEDRS= parameter has been removed.
 - The MBLOC=, MBSIZE=, and INFOPRE= options have been added.
 - The default for the parameter OCPTRTM= has been changed from TMSG to NO.
- ◆ The task detach exit (TDUE=) is not driven if the task is not signed on to SUPRA at detach.
- ◆ PDM information transactions ISRA, ISRB, ISRC, ISRD, ISRE, and ISRF have been added.
- ◆ The CSTXGNIT sample program is included for security system time-out exits.
- ◆ The automatic disconnect force= default has been changed from AUTO to SINF.
- ◆ The C\$TXUNCP macro has been added. This macro generates a list of CICS transactions that are not subject to RSSO inflight processing. As shipped it includes CEDF and CSKP.

Conventions

The following table describes the conventions used in this document series:

Convention	Description	Example
Constant width type	Represents screen images and segments of code.	<pre>PUT 'customer.dat' GET 'miller\customer.dat' PUT '\DEV\RMT0'</pre>
Slashed b (<i>b</i>)	<p>Indicates a space (blank).</p> <p>The example indicates that four spaces appear between the keywords.</p>	<pre>BEGNbbbbSERIAL</pre>
Brackets []	<p>Indicate optional selection of parameters. (Do not attempt to enter brackets or to stack parameters.) Brackets indicate one of the following situations:</p> <p>A single item enclosed by brackets indicates that the item is optional and can be omitted.</p> <p>The example indicates that you can optionally enter a WHERE clause.</p> <p>Stacked items enclosed by brackets represent optional alternatives, one of which can be selected.</p> <p>The example indicates that you can optionally enter either WAIT or NOWAIT. (WAIT is underlined to signify that it is the default.)</p>	<pre>[WHERE <i>search-condition</i>]</pre> <pre><u>(WAIT)</u> (NOWAIT)</pre>
Braces { }	<p>Indicate selection of parameters. (Do not attempt to enter braces or to stack parameters.) Braces surrounding stacked items represent alternatives, one of which you must select.</p> <p>The example indicates that you must enter ON or OFF when using the MONITOR statement.</p>	<pre>MONITOR {ON } {OFF }</pre>

Convention	Description	Example
Underlining (In syntax)	Indicates the default value supplied when you omit a parameter. The example indicates that if you do not choose a parameter, the system defaults to WAIT. Underlining also indicates an allowable abbreviation or the shortest truncation allowed. The example indicates that you can enter either STAT or STATISTICS.	<code>(WAIT)</code> <code>(NOWAIT)</code> <code>STATISTICS</code>
Ellipsis points...	Indicate that the preceding item can be repeated. The example indicates that you can enter multiple host variables and associated indicator variables.	<code>INTO :host-variable [:ind-variable],...</code>
UPPERCASE lowercase	In most operating environments, keywords are not case-sensitive, and they are represented in uppercase. You can enter them in either uppercase or lowercase.	<code>COPY MY_DATA.SEQ</code> <code>HOLD_DATA.SEQ</code>
<i>Italics</i>	Indicate variables you replace with a value, a column name, a file name, and so on. The example indicates that you must substitute the name of a table.	<code>FROM table-name</code>
Punctuation marks	Indicate required syntax that you must code exactly as presented. () parentheses . period , comma : colon ' ' single quotation marks	<code>(user-id, password, db-name)</code> <code>INFILE 'Cust.Memo' CONTROL</code> <code>LEN4</code>
SMALL CAPS	Represent a required keystroke. Multiple keystrokes are hyphenated.	ALT-TAB
<code>OS/390</code> <code>VSE</code>	Information specific to a certain operating system is flagged by a symbol in a shadowed box (<code>OS/390</code>) indicating which operating system is being discussed. Skip any information that does not pertain to your environment.	<code>OS/390</code> See the SUPRA Server procedure library member TIS\$RDM for a list of RDM procedures. <code>VSE</code> See the SUPRA Server RDM sublibrary member TXJ\$INDX for a list of JCL.

SUPRA Server documentation series

SUPRA Server is the advanced relational database management system for high-volume, update-oriented production processing. A number of tools are available with SUPRA Server including DBA Functions, DBAID, precompilers, SPECTRA, and MANTIS. The following list shows the manuals and tools used to fulfill the data management and retrieval requirements for various tasks. Some of these tools are optional. Therefore, you may not have all the manuals listed. For a brief synopsis of each manual, refer to the *SUPRA Server Digest (OS/390 & VSE)*, P26-9062.

Overview

- ◆ *SUPRA Server PDM Digest (OS/390 & VSE)*, P26-9062

Getting started

- ◆ *SUPRA Server PDM Migration Guide (OS/390 & VSE)*, P26-0550*
- ◆ *SUPRA Server PDM CICS Connector Systems Programming Guide (OS/390 & VSE)*, P26-7452

General use

- ◆ *SUPRA Server PDM Glossary*, P26-0675
- ◆ *SUPRA Server PDM Messages and Codes Reference Manual (RDM/PDM Support for OS/390 & VSE)*, P26-0126

Database administration tasks

- ◆ *SUPRA Server PDM and Directory Administration Guide (OS/390 & VSE)*, P26-2250
- ◆ *SUPRA Server PDM Directory Online User's Guide (OS/390 & VSE)*, P26-1260
- ◆ *SUPRA Server PDM Directory Batch User's Guide (OS/390 & VSE)*, P26-1261
- ◆ *SUPRA Server PDM DBA Utilities User's Guide (OS/390 & VSE)*, P26-6260
- ◆ *SUPRA Server PDM Logging and Recovery (OS/390 & VSE)*, P26-2223
- ◆ *SUPRA Server PDM Tuning Guide (OS/390 & VSE)*, P26-0225
- ◆ *SUPRA Server PDM RDM Administration Guide (OS/390 & VSE)*, P26-8220
- ◆ *SUPRA Server PDM RDM PDM Support Supplement (OS/390 & VSE)*, P26-8221
- ◆ *SUPRA Server PDM RDM VSAM Support Supplement (OS/390 & VSE)*, P26-8222
- ◆ *SUPRA Server PDM Migration Guide (OS/390 & VSE)*, P26-0550*
- ◆ *SUPRA Server PDM Windows Client Support User's Guide*, P26-7500*
- ◆ *SPECTRA Administrator's Guide*, P26-9220

Application programming tasks

- ◆ *SUPRA Server PDM DML Programming Guide (OS/390 & VSE)*, P26-4340
- ◆ *SUPRA Server PDM RDM COBOL Programming Guide (OS/390 & VSE)*, P26-8330
- ◆ *SUPRA Server PDM RDM PL/1 Programming Guide (OS/390 & VSE)*, P26-8331
- ◆ *SUPRA Server PDM Migration Guide (OS/390 & VSE)*, P26-0550*
- ◆ *SUPRA Server PDM Windows Client Support User's Guide*, P26-7500*

Report tasks

- ◆ *SPECTRA User's Guide*, P26-9561



Manuals marked with an asterisk (*) are listed more than once because you use them for multiple tasks.



Educational material is available from your regional Cincom education department.

1

Understanding the CICS Connector

The Customer Information Control System (CICS) provides teleprocessing services for application programs, SUPRA Server components, and other Cincom products, including Directory Maintenance, Relational Data Manager (RDM), SUPRA Software Selection Facility, SPECTRA, and MANTIS. Application programs and SUPRA Server components communicate with the Physical Data Manager (PDM) under control of the CICS Connector. When the CICS Connector connects to the PDM (Physical Data Manager), CICS tasks may sign on to the PDM and access the database. The Connector, therefore, makes an association between a PDM task and a CICS task.

To understand the CICS Connector, you need to know:

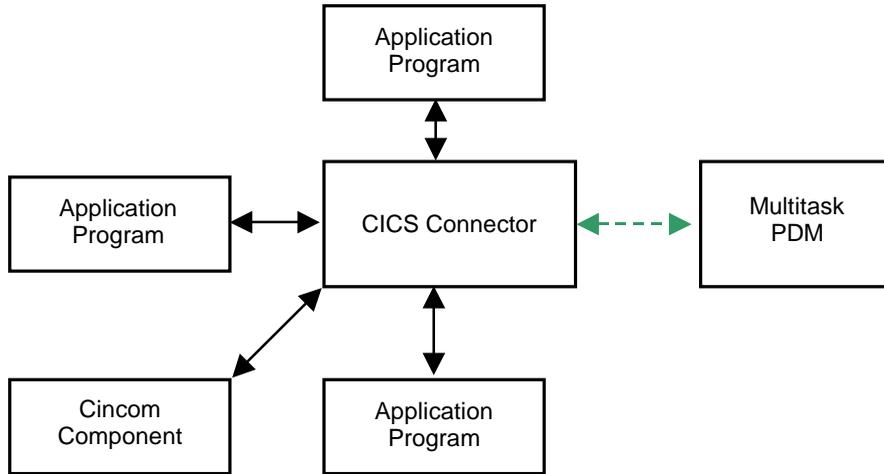
- ◆ The operation of the CICS Connector, including the definition of CICS tasks and transactions. See “[How the CICS Connector works](#)” on page 18 for additional information.”
- ◆ The operation of SUPRA Server components and other Cincom products with the CICS Connector, including the Relational Data Manager (RDM), SPECTRA, MANTIS, and Directory Maintenance (See “[How SUPRA Server components and related products work with CICS and the CICS Connector](#)” on page 34)
- ◆ Physical Data Manager (PDM) operating modes (See “[How PDM operating modes work with CICS](#)” on page 41)

For application programming information, refer to:

- ◆ *SUPRA Server PDM RDM COBOL Programming Guide (OS/390 & VSE)*, P26-8330
- ◆ *SUPRA Server PDM RDM PL/1 Programming Guide (OS/390 & VSE)*, P26-8331

How the CICS Connector works

All application programs running under one copy of CICS communicate with the same CICS Connector, which then communicates with the multitask PDM (see the following figure). A CICS task runs under the control of CICS.



LEGEND

- ↔ Program control transfer. (DFHRMCAL)
- ⇄ Indirect transfer of control.

It is important to understand the distinction between CICS tasks and CICS transactions. CICS tasks execute programs to process transactions. A CICS transaction is work organized as a series of distinct data exchanges to achieve a specific objective. The following sections further explain the relationship between programs, tasks, and transactions.

CICS tasks and the CICS Connector

A SUPRA Server task is a unit of work performed to process one or more transactions. A task may consist of one or several logical units of work. A logical unit of work is a group of database requests, which must all be completed together to perform a complete processing function (the processing executed between COMMIT commands).

The size of a logical unit of work depends on whether task logging is active. If task logging is active, all updates and relevant information are written to a Task Log File for each logical unit of work of each task. Since a logical unit of work is the processing executed between COMMIT commands, a restart in a task logging environment involves reexecuting from the most recent COMMIT.

If task logging is not active, a logical unit of work is the processing executed between PDML SINON and SINOF. Therefore, a restart when task logging is not active involves reexecuting your entire program.

For more information on task logging, the Task Log File, and logical units of work, refer to:

- ◆ *SUPRA Server PDM and Directory Administration Guide (OS/390 & VSE)*, P26-2250
- ◆ *SUPRA Server PDM Logging and Recovery Guide (OS/390 & VSE)*, P26-2223

The CICS Connector supports multiple CICS tasks concurrently. To provide this support, the CICS Connector must keep track of how many tasks will be active and how many functions will be executing at any one time through the Connector. The CONNECT operator control command supplies this information.

A CICS task may be associated with a PDM task from PDM SINON to SINOF of the task via the services of the CICS Connector. A PDM task may be associated with more than one CICS task during the life of a CICS transaction, but never more than one CICS task concurrently.



Multiple CICS tasks can be processing against the PDM, but the statement above means that logically there is a one-to-one correspondence between an active CICS task and a signed-on PDM task.

Special System CICS Connector functions may violate this one-task rule and perform critical operations on a PDM task associated with another CICS task or no CICS task. For example, RSSO (Reset/Sign-Off Processing) running as a nonterminal task may sign off a PDM task associated with another failed CICS task. The correspondence is maintained via the terminal ID associated with the task or, if a nonterminal task, with the CICS task ID.

In a multitask PDM environment, there may be PDM tasks not associated with the CICS Connector interface ID that is currently connected.

A CICS task is usually associated with one terminal. Each task owns resources such as memory and data. Each task signs on to the multitask PDM and owns resources within the PDM.

The standard ID used to identify tasks to the CICS Connector and to the PDM component, if the PDM component ID has not been modified by the TUID user exit (see “[The CSTXUDAT user exit](#)” on page 107), is constructed as follows:

Terminal-related task:	ID
Byte 1:	T (Terminal-related task)
Bytes 2 to 5:	4-byte terminal name from the CICS EIBTRMID field
Bytes 6 to 8:	X'404040'
Nonterminal task:	ID
Byte 1:	N (Nonterminal task)
Bytes 2 to 6:	Decimal display conversion of the task number from the CICS EIBTASKN field
Bytes 7 to 8:	X'4040'

CICS transactions and the CICS Connector

The CICS Connector supports three types of transactions:

- ◆ conversational
- ◆ pseudoconversational
- ◆ nonconversational

How the application programs making up a transaction are written determines the transaction type.

Conversational transactions

A conversational transaction involves several inputs and outputs. A conversation is held between the terminal and the CICS application program. The program and its associated context memory areas remain in main memory for the duration of the transaction, including the time the program is waiting for a response from the terminal.



If you initiate a conversational transaction before the CICS Connector is initialized, the transaction receives a NOTO status on all PDML commands. You must terminate the transaction and then reinitiate it after the completion of the CONNECT operator control command.

Pseudoconversational transactions

A pseudoconversational transaction is similar to a conversational transaction except the CICS task's associated context memory areas do not remain in virtual storage after the output has been written to the terminal. This is because the task is detached at this time. After another input is received from the terminal, the task's context areas are restored via the creation of a new task.

For example, assume a CICS task, TASK1, signs on to the PDM creating a PDM task associated with the CICS Connector currently servicing that transaction. TASK1 is detached after the terminal output is complete. Terminal input causes a new CICS task, TASK2, to be scheduled to continue handling the same transaction. Only one PDM task associated with the CICS Connector has been active and servicing the transaction. It was associated with CICS TASK1, then no CICS task during terminal I/O wait, and finally, with CICS TASK2.

Pseudoconversational transactions must commit PDM data between CICS tasks and return to CICS specifying the next transaction ID to be handled correctly by the CICS Connector.

Nonconversational transactions

A nonconversational transaction consists of accepting input from a terminal, initiating the transaction, sending a reply to the terminal, and terminating the transaction. Nonconversational transactions must SINON and SINOF the PDM before the CICS task returns control to CICS and detaches.



Warning: If you use a BMS page command with the Release option, the CICS task will return control to CICS and detach.

Connecting and disconnecting the PDM

For a CICS task to successfully communicate with the PDM, you must connect the CICS Connector to the PDM by issuing the CONNECT operator control command (See “CONNECT command” on page 87). The illustration at the end of this section shows the transfer of control between CICS and the PDM. You can install the Connector to automatically connect to the PDM at CICS initialization and/or to automatically disconnect at normal CICS termination (CEMT P SHUT). See “Program List Table (PLT)” on page 68 for information on how to do this.

You can also issue the DISCONNECT operator control command to detach the CICS Connector. See “DISCONNECT command” on page 91 for details of the DISCONNECT command. When CICS terminates (either normally or abnormally), the PDM does not terminate, unless it is running in attached mode under CICS.

The OPER CONNECT command creates a SUPRA CICS Connector environment running within your CICS address space that can handle DATBAS requests from your applications or other Cincom components. This environment also provides you with exits and system support functions (task detach management and CICS synchronization). The OPER CONNECT command creates the environment and allocates necessary resources. The OPER DISCONNECT command removes the environment and frees resources.

- ◆ OPER CONNECT can be activated in three ways:
 - Direct entry from a terminal.
 - From a Program List Table (PLT) reference to CSTXPLCI. This action establishes the SUPRA CICS Connector environment automatically during CICS initialization.
 - As terminal input to a simulated BSAM sequential terminal.



BSAM sequential terminal is the recommended method.

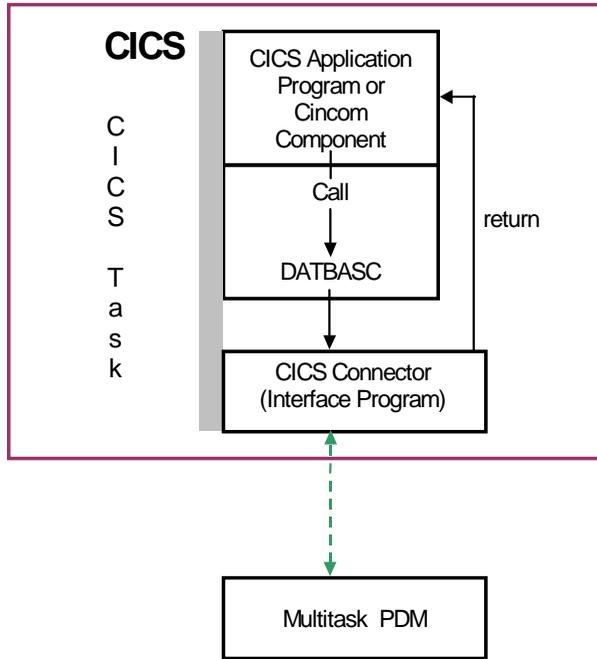
- ◆ OPER DISCONNECT can be activated in three ways:
 - Direct entry from a terminal.
 - From a Program List Table (PLT) reference to CSTXPLDT. This option removes the SUPRA CICS Connector environment and releases resources automatically during CICS normal termination processing (CEMT P SHUT).
 - From a terminal with the FORCE=AUTO or FORCE=SINF option. These options start a nonterminal task to manage the DISCONNECT, thus freeing up the terminal.



You must have an XLT table entry for the OPER transaction code. Refer to the DFHXLT sample on the OS/390 CICSTBLS library or the DFHXLT.TABLE member on the VSE library.

For more information on connecting and disconnecting the CICS Connector to and from the PDM, see [“Connecting the CICS Connector to the PDM”](#) on page 46 and [“Disconnecting the CICS Connector from the PDM”](#) on page 47.

The following figure illustrates task flow to the PDM.



LEGEND

- ◄—► Program Control transfer.
- ◄- - -► Indirect transfer of control.

How the CICS Connector allocates storage

The SUPRA CICS Connector allocates most of its initial memory during the connect process. Some additional memory storage may be obtained during processing to handle requirements unique to task or system load increases. (See the MEME parameter on the [CSTXOPRM macro](#) and the discussion of the CONNECT command in this chapter for more information.) The memory may be obtained from above or below the 16 MB line.



For information on estimating your memory storage requirements, refer to the [SUPRA Server OS/390 Installation Guide](#), P26-0149, or the [SUPRA Server VSE Installation Guide](#), P26-0132.

The memory storage is owned by the Task Related User Exit (TRUE). Memory will be retrieved from CICS DSA.

The memory required depends on the number of tasks and threads requested. Except as noted below, the memory obtained at OPER CONNECT will not increase. At OPER CONNECT, an initial memory block of 65504 bytes (approximately 64K) is allocated for the CICS Connector. During connect processing, additional 65504-byte memory blocks are allocated as needed. The free, unused storage at any time is therefore always less than 65504 bytes, since the last memory block allocated contains whatever data will not fit into the existing memory blocks. The connector memory may reside in extended storage.

Additional memory blocks may be requested by the connector for the following:

- ◆ Cincom component tasks. The number of such tasks is usually small relative to the number of user tasks issuing DATBAS requests. However, several Cincom components require special Directory support physical DML (DIMOPS) to manage user requests, and these may require additional storage.
- ◆ Tasks issuing RDML. These may cause additional memory usage because View Open commands require special Directory support physical DML (DIMOPS) to collect the information needed to open the view.
- ◆ In a VSE environment, allocation of XPCC packets is the largest potential source of memory requests. Refer to the [MAXPACK parameter](#) on the CSTXOPRM macro for more information on XPCC packets.

You can indicate the maximum number of additional 65504-byte memory blocks available to the SUPRA CICS Connector in the CSTXOPRM macro. See the [MEME parameter](#) on the CSTXOPRM macro for more information on this function. These memory blocks are freed at OPER DISCONNECT. See [“Using Cincom CICS Connector CSTXOPRM macro”](#) on page 135 for information about the CSTXOPRM macro.

In addition to the memory managed in the memory blocks allocated at OPER CONNECT, direct requests for dynamic CICS memory occur to handle ongoing memory requirements during processing. With the exception of the global area associated with the Task Related User Exits (TRUEs) enabled by the SUPRA CICS Connector, all additional memory is associated with CICS tasks. This dynamic CICS memory is allocated as follows:

- ◆ Approximately 3K is allocated for the global area associated with the Task Related User Exits, which persists during the entire time the CICS Connector is activated.
- ◆ Approximately 1K (900 bytes) is allocated for each CICS task when it is started. This memory is freed when the task detaches.
- ◆ An additional 4K is allocated for a physical CICS task that actually issues DATBAS requests. This memory is freed when the task detaches.

The total amount of this dynamic memory allocated outside of the 65504-byte memory blocks at any one time depends on the number of active and suspended CICS tasks, and the number of those tasks actually issuing DATBAS requests.

How the CICS Connector handles task recovery

The CICS Connector provides task recovery in order to maintain database integrity and to ensure an operational environment. Task recovery is provided when CICS tasks associated with SUPRA PDM tasks end (either normally or abnormally), or if conditions around an existing SUPRA PDM task change, even if the CICS task is not currently related to that SUPRA PDM task.

A CICS task becomes related to a SUPRA PDM task when the program run by the CICS task successfully issues a DATBAS request that performs a SINON PDML. (A temporary association also exists between a CICS task issuing a DATBAS request and the SUPRA CICS Connector for the duration of the PDML request, even if there is no signed-on SUPRA PDM task.) The relationship is mapped through the task ID. If the task ID is also the associated terminal ID, the SUPRA PDM task is related to that logical terminal ID, and can thus be related to several CICS tasks. See “[CICS tasks and the CICS Connector](#)” on page 19 for more information on the relationship between CICS tasks and the CICS Connector.

A CICS task can become related to an existing SUPRA PDM task, even though the current program associated with the CICS task has not issued any PDML, provided a prior task issued a successful sign-on and then returned control to CICS by passing a “next” transaction ID. See the discussion of pseudoconversational transactions in “[CICS transactions and the CICS Connector](#)” on page 21.

After the CICS task associated with a pseudoconversational transaction detaches, it is possible that the associated terminal may go out of service (because of a failure in a VTAM network). To handle this situation, the CICS Connector provides a CICS VTAM Node Error Program sample, known as the CICS Connector NEP.

OS/390

Refer to the DFHZNEP member in the SUPRA OS/390 MACLIB for more information.

VSE

Refer to the DFHZNEP member in the SUPRA VSE library.

The CICS Connector NEP starts the CSTXLST1 program (LST1 transaction). CSTXLST1 can search a table and start any additional transactions you specify to locate SUPRA PDM tasks related to the terminal in CICS application-owning regions.

For information on tailoring the CICS Connector CSTXLST1 program for MRO, see [“Tailoring the CICS Connector node error program”](#) on page 76.

There are several ways a CICS task can end. The recovery actions the CICS Connector may take are as follows:

- ◆ CICS task returns and detaches normally after a PDM task sign-off.

Recovery action: None.

- ◆ CICS task returns and detaches normally, but did not issue a PDM task sign-off.

- PDM task has uncommitted updates.

Recovery action: Reset and sign off the PDM task.

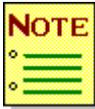
- PDM task does not have uncommitted updates.

Recovery action: If the CICS task passed a “next” transaction ID, no action is necessary. If the CICS task did not pass a “next” transaction ID, sign off the PDM task.

- ◆ CICS task abends and was not actively processing a physical DML (PDML) request.

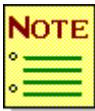
Recovery action: If the CICS task returned and detached normally after PDM task sign-off, no action is necessary. If the CICS task did not sign off, reset and sign off the PDM task, regardless of whether it has uncommitted updates or not.

- ◆ CICS task that was waiting for a PDML request to complete detaches.



This condition usually occurs when you cancel a CICS task, but it can also occur for other reasons, such as MRO remote communication problems.

Recovery action: At the time of CICS task detach, the CICS Connector will try to complete the PDML a specific number of times at one-second intervals (you specify the number on the CSTXOPRM macro under IFRTRY=*n*). The CICS Connector will continue to retry until *n* is reached or until the ECB used for the original wait is posted by the PDM.



IFRTRY=*n* needs to be greater than the TP-MONITOR DELAY TIME parameter for the currently active PDM environment descriptor.

If *n* is reached, the recovery is abandoned, and future PDML requests from CICS tasks on this terminal will receive ASOT status.

If the ECB is posted, the interface side of the PDML completes. If the PDML was a sign-off, no further action is necessary, otherwise reset and sign-off the PDM task.

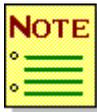
You can modify the recovery actions discussed earlier by using the Task Detach User Exit (TDUE) option on the CSTXOPRM macro to define user exit processing. For more information on the TDUE exit, see “[The TDUE \(Task Detach\) user exit](#)” on page 124.

You can use the CSTXOPRM OCPTRTM= option to start a CICS transaction at the end of RSSO processing. The default OCPTRTN= is NO which means that no transaction is started. A sample program (CSTXTMSG) is supplied in source that can be used to send a message to the terminal operator at the end of RSSO processing.

How to retry PDML

A physical DML (PDML) may return one of three retryable status codes—CFUL, ICOR, and TFUL—because of a temporary lack of a particular resource.

- ◆ CFUL means that the number of tasks specified on the OPER CONNECT command are currently in use. However, if any of these tasks sign off and remove the SUPRA PDM task, that task's resources are then free and available for use.
- ◆ TFUL means that the number of threads specified on the OPER CONNECT command are currently in use. When a PDML in progress completes, a thread is cleared and available for the new task to use.
- ◆ ICOR means that there is insufficient memory to service the PDML at the time of the request. Again, when some of the active tasks complete, enough storage may be freed to allow the new task to process its PDML.



Warning: ICOR conditions generally mean that you are exceeding your resource allocations. You will probably not be able to continue processing for much longer without reconfiguring.

Because the missing resource indicated by the above three status codes may become available, you can specify the number of retries for all three both in the CSTXOPRM macro and with the OPER CONNECT command. You can specify the number of one-second delays that will be attempted before the CFUL, TFUL, or ICOR status is actually returned to the application.

The default values are TFUL=30 (50 in the sample CSTXOTBL), CFUL=30 (50 in the sample CSTXOTBL), and ICOR=5 (20 in the sample CSTXOTBL). The number of ICOR retries is smaller because you are exceeding your resource allocations.

If you wish, you can increase these retry counters so that attempting to use more threads or tasks than specified at OPER CONNECT will not cause tasks to fail as a result of CFUL or TFUL statuses being returned. However, the response time will be longer when the system load exceeds the average number of tasks and threads available, because of the increased number of retries.

Conversely, you may decrease the number of retries to gain faster response time (or for debugging or tuning reasons), but more tasks could fail as a result.

How operator commands are processed

You process operator commands by invoking the Operator Control Task. The Operator Control Task processes a single operator control command. The Operator Control Task may be invoked either from a 3270-type terminal or from the operator console (if the operator console is defined to CICS). Any number of Operator Control Tasks may be executing at any one time. The CICS Connector Activity Audit Trail (“[CICS Connector Activity Audit Trail](#)” on page 56) documents all operator control activity.

The Operator Control Task processes the following:

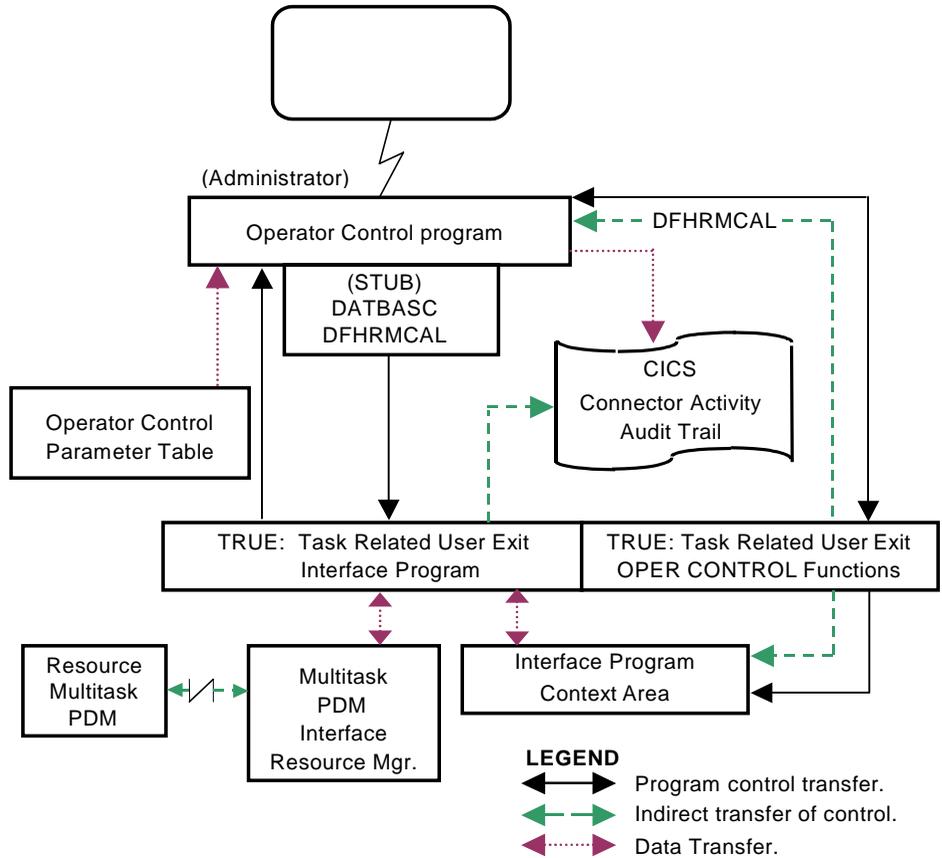
- ◆ The CONNECT operator control command (“[CONNECT command](#)” on page 87)
- ◆ The STATUS operator control command, by displaying the CICS Connector status on the terminal (“[STATUS command](#)” on page 98)
- ◆ The DISCONNECT, DUMP, and ABEND operator control commands (“[DISCONNECT command](#)” on page 91, “[DUMP command](#)” on page 93, and “[ABEND command](#)” on page 84)
- ◆ The OPEN and CLOSE operator control commands, by acting as a SUPRA Server application program (“[OPEN command](#)” on page 94 and “[CLOSE command](#)” on page 85)
- ◆ The PURGE command, to reset and sign off specific PDM tasks (“[PURGE command](#)” on page 96)

If you specify defaults for the operator control commands, they are extracted from the Operator Control Parameter Table (“[Generating the operator control parameter table \(CSTXOTBL\)](#)” on page 72). This table is dynamically loaded by a CICS load request. You can tailor this table to reflect your normal run time parameters.



A Reset/Sign-off (RSSO) operator control command is internally generated to provide RESET and SINOF operations for PDM tasks in various error states. The RSSO flag in the Signed On Table (SOTB) must have been set when a detach was detected. The Reset/Sign-off (RSSO) function is also automatically scheduled. When RSSO runs, it searches the allocated SOTB entries and processes only those flagged. If you schedule RSSO from the terminal, it probably will not find any SOTB entries with the RSSO flag on, and it is therefore a null operation. Use OPER PURGE instead.

The following figure illustrates Operator Control Task processing.



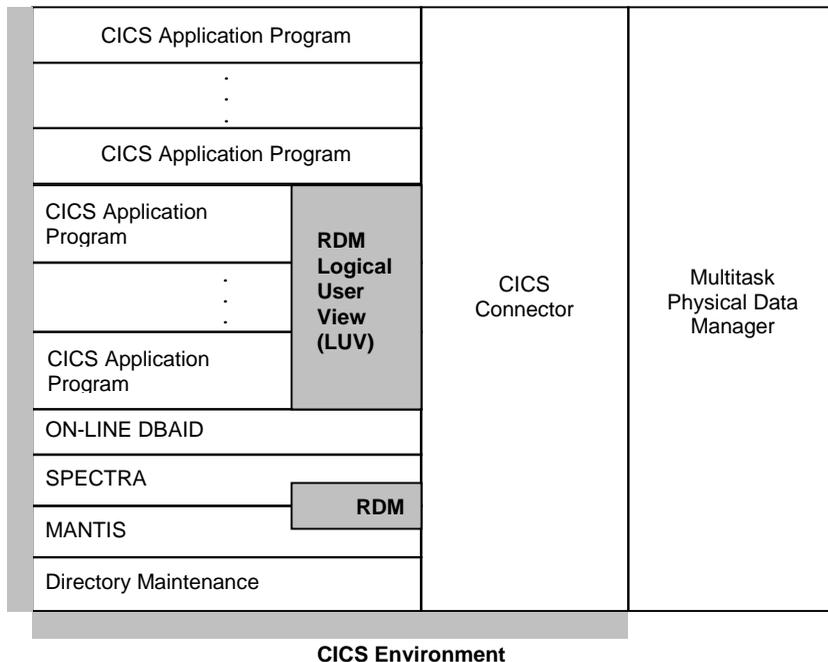
How SUPRA Server components and related products work with CICS and the CICS Connector

The SUPRA Server components and related products (RDM, SPECTRA, MANTIS, and Directory Maintenance) communicate with the PDM through the CICS Connector. CICS application programs, MANTIS, and SPECTRA may communicate with either the Relational Data Manager (RDM) using RDML commands or with the CICS Connector directly using physical DML (PDML) commands. RDM and Directory Maintenance communicate directly with the CICS Connector. The CICS Connector communicates with the PDM.

Different tasks can execute the same CICS application program. CICS builds the environment in which SUPRA operates and the CICS Connector exists.

For specific tuning information, refer to the *SUPRA Server PDM Tuning Guide (OS/390 & VSE)*, P26-0225.

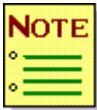
The following figure represents the relationship of the SUPRA Server components and related products to CICS, other SUPRA Server components, and the CICS Connector.



The Relational Data Manager (RDM) and CICS

CICS application programs can communicate with the RDM through the CSVICICS interface module or the CSVNICIC interface module. These interface modules connect the CICS application programs to the RDM interface. The RDM interface manages resources required to service RDM requests and provides all communication to the CICS Connector. RDM uses the following processing sequence:

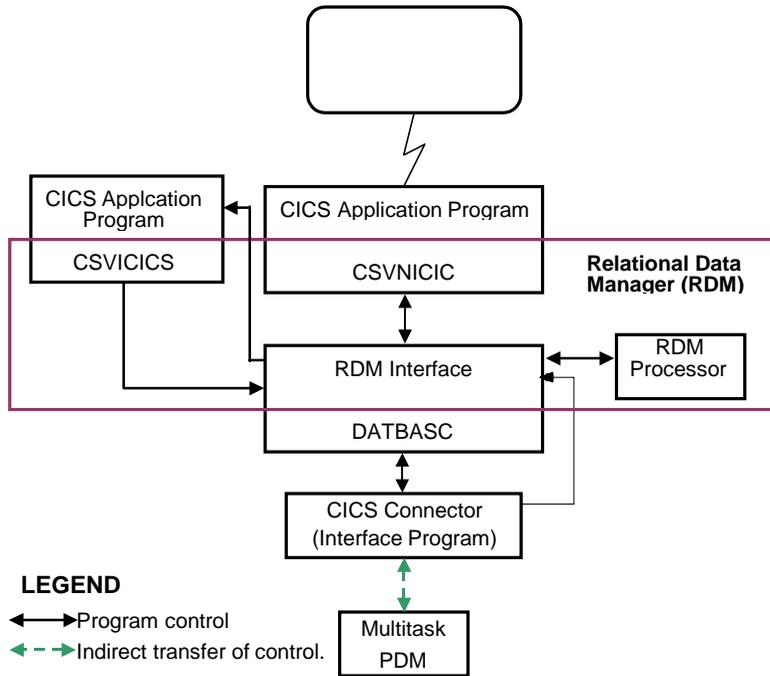
1. The application program passes a request to the RDM interface through the CSVICICS or CSVNICIC interface modules.
2. The RDM interface receives the request and retrieves or allocates RDM memory for the RDM task context. The RDM interface then passes the request to RDM.
3. RDM issues a request to the PDM. This request passes through the RDM interface and is processed through the CICS Connector.
4. The RDM interface returns the serviced PDM request to RDM. When the RDM request is serviced, the RDM interface returns control to the application program.



The CSVICICS module will become obsolete in a future release of SUPRA Server. We recommend that you begin using CSVNICIC when you relink applications. The RDM COBOL and PL/1 preprocessors now generate a reference to CSVNICIC.

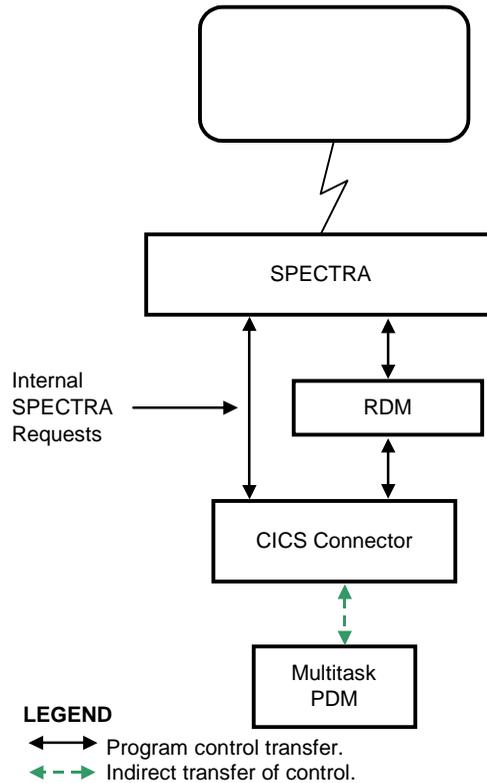
With SUPRA Server Release 2.4 and higher, the CICS RDM resident module will be loaded above the 16 MB line. RDM heaps and global views can also reside above the 16 MB line. Refer to the discussion of the C\$VOOPTM macro in the [SUPRA Server PDM RDM Administration Guide \(OS/390 & VSE\)](#), P26-8220, for more information on placing RDM heaps and global views above the 16 MB line.

The following figure illustrates RDM task processing:



SPECTRA and CICS

SPECTRA functions as a terminal-oriented application task under CICS. Many SPECTRA tasks can run at any one time. Refer to the *SPECTRA User's Guide*, P26-9561, for information on how to use SPECTRA. The following figure shows SPECTRA processing:

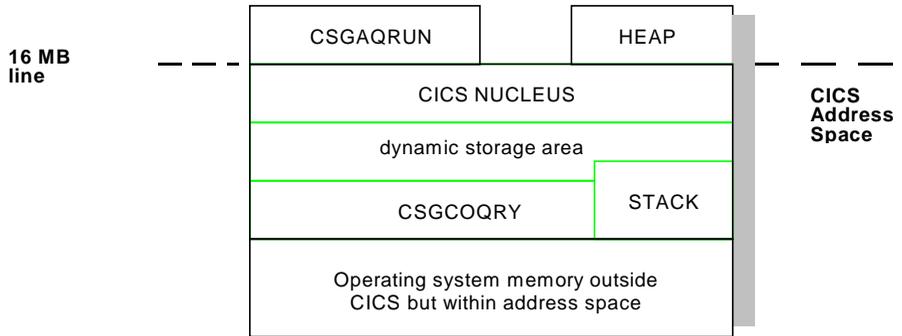


SPECTRA is loaded in the CICS region. You must include the SPECTRA CICS environment module CSGCOQRY in the CICS PPT tables. Refer to the *SPECTRA Administrator's Guide*, P26-9220, for details.

The following figure shows the CICS region and how SPECTRA is configured in an OS/390 environment.

You can place the SPECTRA nucleus, CSGAQRUN, either in the linkpack area or loadlib under OS/390, or into a core image library or the shared virtual area (SVA) under VSE. For SPECTRA context requirements, refer to the *SPECTRA Administrator's Guide*, P26-9220.

With SUPRA Server 2.1.6 and higher, you can load the main SPECTRA module CSGAQRUN above the 16 MB line. If you wish CSGAQRUN to reside above the 16 MB line, refer to the *SPECTRA Administrator's Guide*, P26-9220, for linkedit instructions. RDM heaps and global views can also reside above the 16 MB line. Refer to C\$VOOPTM macro documentation in the *SUPRA Server PDM RDM Administration Guide (OS/390 & VSE)*, P26-8220, for more information on placing RDM heaps and global views above the 16 MB line.

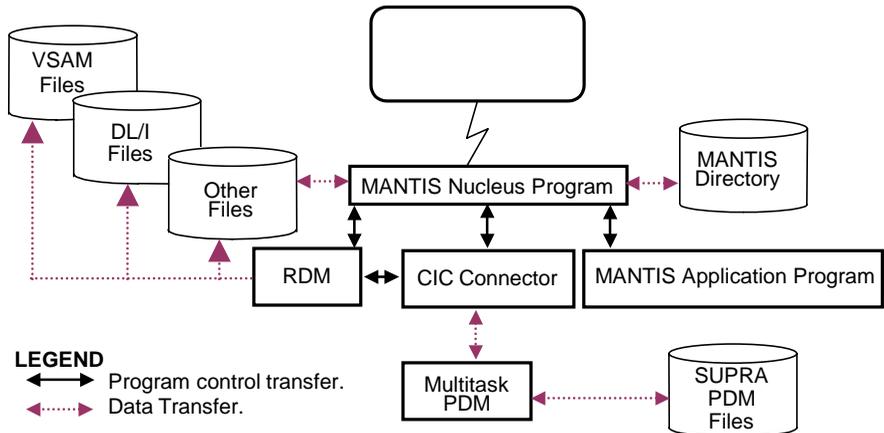


MANTIS and CICS

MANTIS normally functions as a terminal-oriented application task under CICS. Many MANTIS applications can run at any one time. A MANTIS application may contain a logical unit of work which updates the SUPRA Server database files and one or more of the non-SUPRA Server database files available to MANTIS (VSAM, DL/I, etc.) or RDM tasks (VSAM, DL/I). The CICS Connector backs out the SUPRA Server database updates if a CICS task abends.

The MANTIS nucleus interprets and executes programs written in the MANTIS language. MANTIS programs reside in special MANTIS files and require no linking. The MANTIS nucleus is linked when SUPRA Server is installed. Refer to *AD/Advantage MANTIS Administration OS/390, VSE/ESA, P39-5005*, for information on how to write, execute, and tune MANTIS programs.

The following figure illustrates MANTIS application task processing:

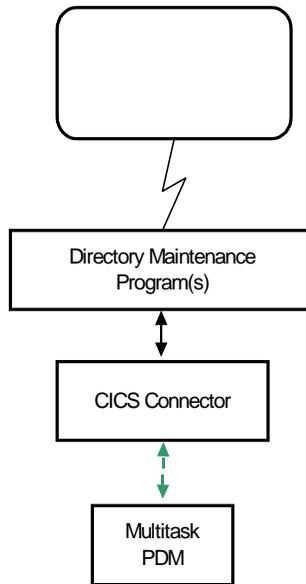


Directory maintenance and CICS

Directory Maintenance functions as a terminal-oriented application task under CICS. Directory Maintenance tasks are conversational. Many Directory Maintenance tasks can run at any one time. Directory Maintenance executes as 2–12 load modules and uses CICS Basic Mapping Support. For information on tuning Directory Maintenance, refer to the *SUPRA Server PDM Tuning Guide (OS/390 & VSE)*, P26-0225.

The Database Administrator (DBA) uses Directory Maintenance to control information on the Directory. The information on the Directory files describes user and system databases, including a description of the Directory database. For more information about Directory Maintenance, refer to the *SUPRA Server PDM and Directory Administration Guide (OS/390 & VSE)*, P26-2250.

The following figure illustrates Directory Maintenance task processing:



LEGEND

- ↔ Program control transfer.
- ◄ - - ► Indirect transfer of control.

How PDM operating modes work with CICS

The PDM can operate with CICS in three modes:

- ◆ attached
- ◆ attached central
- ◆ central

This section provides a general description of these modes. For more specific information on modes of operation, refer to the *SUPRA Server PDM and Directory Administration Guide (OS/390 & VSE)*, P26-2250.

Central operating mode

Central operating mode is the primary operating mode for SUPRA Server 2.1.6 and higher. In the central operating mode, CICS and the PDM normally reside in different address spaces.

VSE

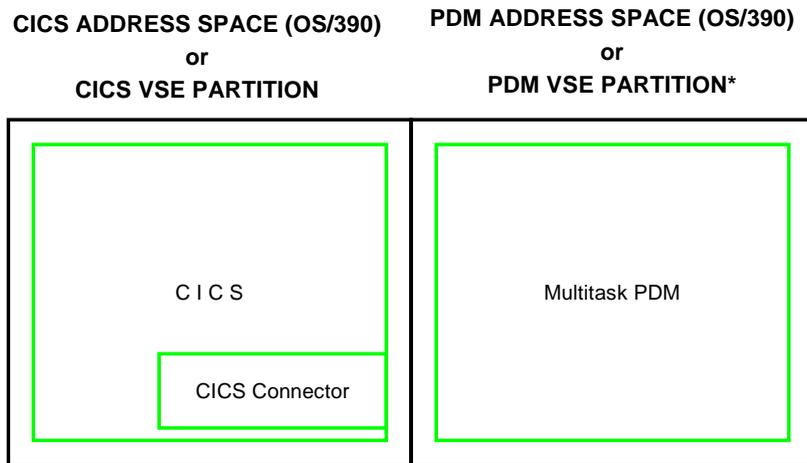
For VSE, CICS, and the PDM can reside either in the same address space but different partitions, or in different partitions in different address spaces.

The PDM can communicate with any program in central mode that signs on to it. If the CICS Connector connects to the PDM, CICS tasks may sign on to the PDM. The PDM can be connected automatically at CICS initialization time or whenever you issue the CONNECT operator control command. See “[CONNECT command](#)” on page 87 for details on the CONNECT command.

Only one copy of the PDM is connected at a time. However, any number of CICS application tasks (communicating through one or more copies of CICS) can access the same copy of the multitask PDM through the CICS Connector.

The central PDM does not have to be in a different address space from CICS. The PDM can be attached to CICS and be the central PDM for the other batch applications or another copy of CICS (see “[Attached central operating mode](#)” on page 44).

The following figure illustrates the CICS central operating mode:



* In the same or different VSE address space as the CICS partition

Attached operating mode

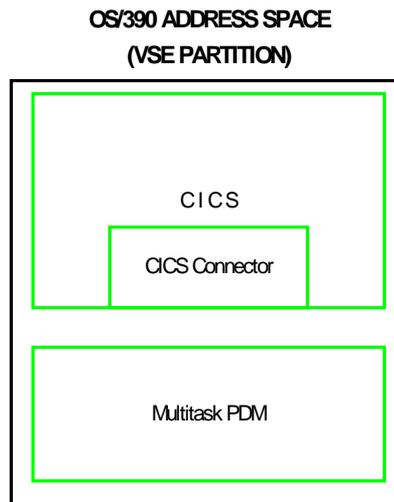
In the attached operating mode, CICS and the PDM reside in the same address space (the same VSE partition), but as separate operating system tasks with the PDM attached under the CICS task. Because CICS acquires all memory of the operating system in this address space, set the CICS run time parameters to return enough memory to the operating system for the PDM to run in.

The PDM can be connected automatically at CICS initialization time or whenever you issue the CONNECT operator control command. See “[CONNECT command](#)” on page 87 for details on the CONNECT operator control command.

Only one copy of the PDM is attached at a time. However, any number of CICS application tasks can access the same copy of the multitask PDM through the CICS Connector.

The PDM terminates and the CICS Connector detaches at normal CICS termination if you installed the automatic disconnect feature. You can also issue the DISCONNECT operator control command to terminate the PDM and detach the CICS Connector. See “[DISCONNECT command](#)” on page 91 for details of the DISCONNECT command. When CICS terminates (either normally or abnormally), the PDM also terminates.

The following figure illustrates the CICS attached operating mode:



Attached central operating mode

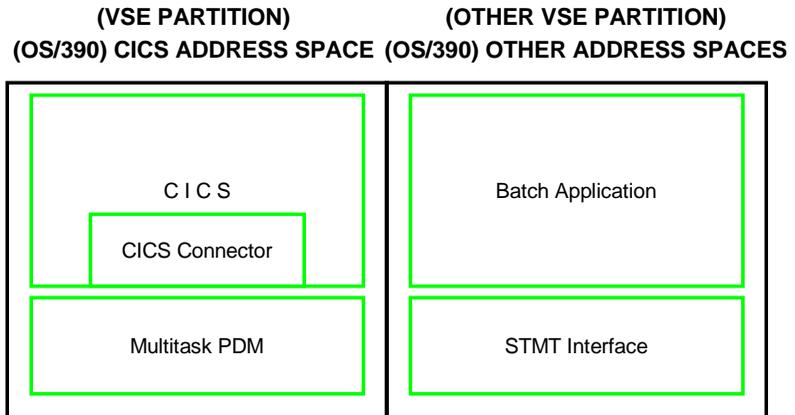
The attached PDM can be accessed as a central PDM by batch application programs or multiple copies of CICS.

In the attached central operating mode, the multitask PDM communicates with the CICS application task through that copy of CICS. The interface associated with that copy of CICS attaches the multitask PDM as an operating system subtask.

Application tasks may reside in different address spaces from the one containing the multitask PDM.

The PDM terminates and the CICS Connector detaches at normal CICS termination if you installed the automatic disconnect feature. You can also issue the DISCONNECT operator control command to terminate the PDM and detach the CICS Connector. See “DISCONNECT command” on page 91 for details of the DISCONNECT command. When CICS terminates (either normally or abnormally), the PDM also terminates regardless of any connections the PDM has with tasks in other address spaces.

The following figure illustrates the CICS attached central operating mode:



2

Operating the CICS Connector

The CICS Connector makes it possible for CICS application programs, SUPRA Server components such as Directory Maintenance and RDM, and other applications such as SPECTRA and MANTIS to communicate with the PDM. You need to understand the following procedures to use the CICS Connector:

- ◆ Connecting the CICS Connector to the PDM
- ◆ Disconnecting the CICS Connector from the PDM
- ◆ Determining the status of the CICS Connector
- ◆ Opening and closing database files dynamically
- ◆ Using the ABEND and DUMP commands
- ◆ Using the PURGE command

Connecting the CICS Connector to the PDM

To connect the CICS Connector to the PDM, issue the CONNECT operator control command (see “CONNECT command” on page 87).



To connect to the PDM, Cincom recommends that you issue the OPER CONNECT command from a terminal defined as a sequential (BSAM or BTAM) device.

OS/390

For more information, consult the *CICS/ESA System Definition Guide* or the *CICS Resource Definition Manual*.

VSE

For more information, consult the BTAM Samples Terminal Control Table in the *CICS Installation and Operations Guide*.

Or you can include an entry in the Program List Table (PLT) to automatically issue the CONNECT when CICS is initialized. During automatic connect, the CONNECT command parameters use default values. You specify the default values for all CONNECT command parameters in the Operator Control Parameter Table. See “Connecting and disconnecting the PDM” on page 23 for more information about automatic connect. For more information about the Program List Table, see “Program List Table (PLT)” on page 68.

You can control the connection of the Connector to the PDM, as well as the execution of the PDM, by coding various parameters in the CSIPARM file. Refer to the *SUPRA Server PDM and Directory Administration Guide (OS/390 & VSE)*, P26-2250, for more information on the CSIPARM file. If you specify the ATTACH parameter, the PDM is attached under CICS as a separate operating system task. If you omit the ATTACH parameter, the CICS Connector searches for a central PDM. Specify the PDM name in the DBM parameter.

While the CICS Connector is connected, CICS may be forced to terminate. Under attached mode, if CICS terminates and a task is signed on, a system abend and an IBM message occur. In this mode, the PDM is an attached subtask and is abnormally terminated. Refer to the appropriate IBM documentation for an explanation of this message. This abend does not occur in a normal unforced termination.

Disconnecting the CICS Connector from the PDM

To disconnect the CICS Connector from the PDM, issue the DISCONNECT operator control command (see “[DISCONNECT command](#)” on page 91). If the CICS Connector is already disconnected from the PDM when you issue a DISCONNECT command, a warning message is returned and the request is ignored. If the PDM is in the central operating mode and you issue a DISCONNECT command, any CICS applications using the PDM will receive a NOTO status. If the PDM is in the attached operating mode, a DISCONNECT command terminates the PDM and then disconnects it.

You can include an entry in the Program List Table (PLT) to automatically issue the DISCONNECT command during CICS termination. (See “[Program List Table \(PLT\)](#)” on page 68 for more information on the PLT.) Automatic DISCONNECT uses the FORCE=SINF option.

The force option on the OPER DISCONNECT FORCE command has the following meanings:

- ◆ FORCE=NO. A DSCNT physical DML command is attempted. If the status is **** (processing normal), disconnect processing continues. If the status is other than ****, the status is reported to the terminal operator and the CICS Connector remains connected to the PDM.
- ◆ FORCE=AUTO. A background task is started which issues DSCNT physical DML commands without the FORCE option:
 - If the status is ACTV and the interface control blocks show active tasks, wait a short time and repeat the DSCNT attempt.
 - If the status is ACTV and no active tasks exist for this interface or status is ****, use FORCE=YES option on DSCNT physical DML command.
 - If status is ****, continue with the DISCONNECT.
- ◆ FORCE=SINF (this is the default). This works the same way as FORCE=AUTO, except that if the status is ACTV and the interface shows active tasks, a RESET and SINOF is attempted for each active task found and the DSCNT physical DML command is attempted again.

If the RESET and SINOF fails for any tasks, a DSCNT physical DML command with force option is attempted.

When you issue the DISCONNECT command, various run-time statistics will print. If you manually issue the DISCONNECT command, these statistics are displayed on your terminal and written on the CICS Connector Activity Audit Trail. If the DISCONNECT command is issued automatically, these statistics are written on the CICS Connector Activity Audit Trail only.

Determining the status of the CICS Connector

To determine whether the CICS Connector is connected or disconnected, issue the STATUS command (see “[STATUS command](#)” on page 98). If the CICS Connector is connected, run-time statistics are written on the CICS Connector Activity Audit Trail and displayed on your terminal. These statistics are reset only when the CICS Connector is disconnected or abends. You can use these statistics to refine the connection parameters. The run-time statistics are:

- ◆ List of the current CSTXOPRM parameters
- ◆ Number of tasks signed on to the PDM
- ◆ Number of CFUL statuses automatically retried
- ◆ Number of CFUL statuses returned to the application
- ◆ Number of TFUL statuses automatically retried
- ◆ Number of TFUL statuses returned to the application
- ◆ Number of ICOR statuses automatically retried
- ◆ Number of ICOR statuses returned to the application
- ◆ Memory in use versus memory free

If you issue the STATUS command while the CICS Connector is disconnected, you receive a message saying the connector is not currently connected. See “[How to retry PDML](#)” on page 31 for additional information about the status codes listed above. All status codes are documented in the [SUPRA Server PDM Messages and Codes Reference Manual \(RDM/PDM Support for OS/390 & VSE\)](#), P26-0126.

Opening and closing database files dynamically

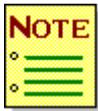
To open and close PDM database files dynamically, issue the OPEN and CLOSE operator control commands (see “[OPEN command](#)” on page 94 and “[CLOSE command](#)” on page 85). With one command, you can open and close individual files or a group of files. You can define groups of files in the Operator Control Parameter Table. Each group must contain at least one file. However, a file may appear in more than one group. The maximum number of files in a group is determined by how much main memory is available to hold control blocks for the OPEN and CLOSE commands. Approximately 12 bytes of memory are used for each file. See “[Generating the operator control parameter table \(CSTXOTBL\)](#)” on page 72 for more information about the Operator Control Parameter Table.

For information on calculating memory requirements, refer to the [SUPRA Server PDM Tuning Guide \(OS/390 & VSE\)](#), P26-0225. Additional information on memory requirements can be found in the [SUPRA Server OS/390 Installation Guide](#), P26-0149, or the [SUPRA Server VSE Installation Guide](#), P26-0132.

If an error occurs while opening or closing any file in a group, all files in the group are returned to their original status. Because of this, you should make sure when defining a file as belonging to more than one group that both groups will not be opened at the same time. The name of the file receiving the error is displayed on the terminal with an error message. All OPEN and CLOSE commands and resulting messages are written on the CICS Connector Activity Audit Trail.

Consider the following exception conditions before opening and closing files:

- ◆ Do not use the OPEN and CLOSE commands on the Directory files. All five Directory files are treated as a logical group and should not be opened or closed individually. This ensures the integrity of the Directory files.
- ◆ If you open files explicitly for CICS processing and then close them for processing by a batch or other PDM task, any recovery on those files using the CICS system logs must be completed before another PDM updates the file.
- ◆ Nonterminal tasks should never open files with EUPD (exclusive update). If the CICS system or CICS Connector abends after a CICS nonterminal task opens a database file with EUPD, that file will not be available on a warm reconnect.



Cincom recommends that no CICS task opens files with EUPD.

Purging a task

The PURGE command allows you to purge a SUPRA PDM task associated with one terminal from another terminal. The PURGE command frees the resources in the CICS Connector associated with the task.

To purge a task, enter OPER PURGE ID= taskid at another terminal. This procedure schedules the RSSO transaction for the task.



The PURGE command is automatically generated when the CICS Connector Node Error Program (NEP) recovers context for a task associated with a failed terminal.

3

Resolving problems

Problems or unforeseen situations can occur in both SUPRA Server and CICS. These problems frequently appear during normal operation and generate error messages. In addition, problems can occur in the SUPRA CICS system because you have set up closely related parameters in separate Cincom components in a conflicting manner. Debugging tools help you identify these problems, their locations, and their resolutions.

Common SUPRA Server and CICS error messages and symptoms

The following are common error messages and symptoms, with suggested solutions to the problems they may represent. Other problems may occur that are unrelated to SUPRA Server; however, this discussion is limited to common error messages: 998, Task Dump ID (D998), and the CICS Connector Interface being sealed.

- ◆ *Error Message 0998*—The CICS Connector encountered a severe error causing the interface to cancel the task. Normally, you can find a message describing the error on either the CICS Connector Activity Audit Trail or on the system operator console. Refer to the *SUPRA Server PDM Messages and Codes Reference Manual (RDM/PDM Support for OS/390 & VSE)*, P26-0126, for corrective action. If no message identifies the error, obtain the transaction dump and the dump obtained from the DUMP(or ABEND) command (see “*DUMP command*” on page 93 or “*ABEND command*” on page 84).
- ◆ *CICS Task Dump ID (D998)*—The dump identifier D998 may appear as a CICS dump code when a CICS task abends. This dump identifier appears on dumps caused by the OPER DUMP or OPER ABEND. You can issue the DUMP command without terminating CICS. See “*DUMP command*” on page 93 for instructions on using the DUMP command.



Warning: Do not use codes 0998 and D998 in your programming environment. These codes are reserved for Cincom use only.

- ◆ *CICS Connector Interface Sealed*—You may find that you cannot communicate with the PDM because the CICS Connector interface is sealed (flagged as unusable so that further requests will fail). This can occur if the interface abended or the operator issued an ABEND command. You must recycle CICS before you can reconnect to the PDM.

CICS environment setup problems

Several parameters used in separate Cincom components are closely related, and if these parameters are set up in a conflicting manner, your SUPRA CICS system may not function correctly. Symptoms range from excessive TFUL statuses to being unable to connect to the PDM.

The Cincom parameters also can be affected by the manner in which you have defined your IBM CICS DFHSIT parameters.

The following are changes you can make to Cincom and IBM CICS parameters that are closely related to each other. You should ensure that the values you specify in each case do not conflict with the values in the other parameters:

- ◆ The AD ED (add environment description) Directory Maintenance transaction defines the number of interfaces, tasks, and threads that will be available. Keep in mind that the CICS Connector reserves one task (in addition to the TASKS you specify in CSTXOTBL or in your OPER CONNECT request) for Disconnect processing. Since the PDM ensures that enough tasks are available to service any unconnected interfaces, you will receive an IPAR status if you do not allow for an extra task for each interface that will use CICS.

See “[Using Cincom CICS Connector CSTXOPRM macro](#)” on page 135 for a complete description of these parameters.

- ◆ The CICS Connector table CSTXOTBL (built by the CSTXOPRM macro) specifies tasks and threads. See “[Generating the operator control parameter table \(CSTXOTBL\)](#)” on page 72 for a description of these parameters.
- ◆ For OS/390, the C\$VOOPTM macro defines the number of RDM heaps in your environment. Refer to the [SUPRA Server PDM RDM Administration Guide \(OS/390 & VSE\)](#), P26-8220, for a description of this parameter.

Debugging tools

Debugging tools help you resolve the problems in your system operation. The following debugging tools are useful in resolving the problems discussed earlier:

- ◆ CICS Connector Activity Audit Trail
- ◆ CICS Trace Table Entries
- ◆ ABEND Operator Control Command
- ◆ DUMP Operator Control Command

CICS Connector Activity Audit Trail

The CICS Connector Activity Audit Trail is a log of numbered messages about the status and activity of the CICS Connector. PDM messages and messages from the CSTXCMT program (the PDM/CICS Interface) are written to the system operator console. The following items are written on the log:

- ◆ All operator control commands
- ◆ Responses from the Operator Control Program
- ◆ Messages for connections, disconnections, abends and task recoveries

Refer to the *SUPRA Server PDM Messages and Codes Reference Manual (RDM/PDM Support for OS/390 & VSE)*, P26-0126, for details about these messages.

The audit trail is implemented as a CICS Transient Data Destination and defined in the Destination Control Table (see “[Transient data destination](#)” on page 63) similar to the CSTL and CSCS destinations used by CICS management modules. (The default ID is MUSR. You can change it using the MESSQID parameter on the CSTXOPRM macro.)

You can route this ID directly to a hard-copy printer or indirectly to the CSTL, CSCS or other standard destinations. If you route the ID to an indirect destination ID, that destination ID cannot have DESTRCV=LG specified. If a program reads messages from a destination and displays them on the system operator console, you may route the messages to that destination. You should have the CICS Connector Activity Audit Trail available in a hard copy to aid in diagnosing problems. For information on routing the audit trail to a hard-copy printer, see “[Transient data destination](#)” on page 63.

The following code sample is an example of the CICS Connector Activity Audit Trail:

```

CSTX039I : OPER CONNECT 09:06:18
CSTX053I : THREADS = 00002, TASKS = 00005, TFUL RETRIES = 00050      09:06:18
CSTX053I : CFUL RETRIES = 00050, ICOR RETRIES = 00020, MEME = 00025  09:06:19
CSTX053I : IFRTRY = 00000005, MAXPACK = 00000000, IMMEDRS =        09:06:19
CSTX053I : SYNC = N, TOTC = N, PUSHPOP = N, TRACE1 = 05, TRACE2 = 06 09:06:19
CSTX053I : TRACE3 = 07, MESSQID = MUSR, OCPTRID = OPER, OCPTRM = NO 09:06:19
CSTX053I : USERTID = NO , TDUE = NO , TPOP = NO                    09:06:20
CSTX053I : TPOS = NO , TPRE = NO , TPRP = NO                      09:06:20
CSTX053I : TSYN = NO , TUID = NO                                  09:06:20
CSTX053I :                                                         09:06:20
CSTX048I : THE CICS CONNECTOR HAS BEEN CONNECTED.... 09:06:22
CSTX020I : REQUEST COMPLETED NORMALLY. CLEAR SCREEN TO CONTINUE.... 09:06:22

CSTX039I : OPER STAT 09:07:10
CSTX053I : THREADS = 00002, TASKS = 00005, TFUL RETRIES = 00050      09:07:10
CSTX053I : CFUL RETRIES = 00050, ICOR RETRIES = 00020, MEME = 00025  09:07:10
CSTX053I : IFRTRY = 00000005, MAXPACK = 00000000, IMMEDRS =        09:07:10
CSTX053I : SYNC = N, TOTC = N, PUSHPOP = N, TRACE1 = 05, TRACE2 = 06 09:07:10
CSTX053I : TRACE3 = 07, MESSQID = MUSR, OCPTRID = OPER, OCPTRM = NO 09:07:10
CSTX053I : USERTID = NO , TDUE = NO , TPOP = NO                    09:07:10
CSTX053I : TPOS = NO , TPRE = NO , TPRP = NO                      09:07:10
CSTX053I : TSYN = NO , TUID = NO                                  09:07:10
CSTX053I :                                                         09:07:10
CSTX022I : THE SUPRA CICS CONNECTOR IS CURRENTLY CONNECTED..... 09:07:10
CSTX042I : SIGNED ON TASKS = 00000, PEAK SIGNED ON TASKS = 00001 ... 09:07:10
CSTX043I : CFUL STATS: RETRIES = 0000000000, RETURNED = 0000000000 . 09:07:10
CSTX043I : TFUL STATS: RETRIES = 0000000000, RETURNED = 0000000000 . 09:07:10
CSTX043I : ICOR STATS: RETRIES = 0000000000, RETURNED = 0000000000 . 09:07:10
CSTX080I : MEMORY TOTAL = 000000000065504, FREE = 000000000053136 .. 09:07:10
CSTX020I : REQUEST COMPLETED NORMALLY. CLEAR SCREEN TO CONTINUE.... 09:07:10

CSTX006I : AUTOMATIC TERMINATION OF SUPRA CICS INTERFACE HAS BEEN SCHEDULED. 09:07:38
CSTX039I : DISCONNECT FORCE=SINF 09:07:38
CSTX048I : THE CICS CONNECTOR HAS BEEN DISCONNECTED, FORCE = NO.... 09:07:38
CSTX042I : SIGNED ON TASKS = 00000, PEAK SIGNED ON TASKS = 00001 ... 09:07:38
CSTX043I : CFUL STATS: RETRIES = 0000000000, RETURNED = 0000000000 . 09:07:38
CSTX043I : TFUL STATS: RETRIES = 0000000000, RETURNED = 0000000000 . 09:07:38
CSTX043I : ICOR STATS: RETRIES = 0000000000, RETURNED = 0000000000 . 09:07:38
CSTX020I : REQUEST COMPLETED NORMALLY. CLEAR SCREEN TO CONTINUE.... 09:07:38

```

CICS trace table entries

The CICS Connector places entries in the CICS trace table if the CICS trace facility is active. You can use the trace table to determine the flow of events in an abending task and how those events relate to the CICS Connector. CICS includes the trace table in every CICS transaction dump. The following entries may appear in the trace table:

ID	Data	Definition
005	<i>dddd</i>	The CICS Connector received the PDM DML command <i>dddd</i> for processing.
006	<i>ssss</i>	The CICS Connector returned PDM status <i>ssss</i> to the most recent PDM DML command.
007	TASKID	This is a new pseudoconversational CICS task for the signed-on PDM task ID.

Use the TRACE1, TRACE2, and TRACE3 parameters in the CSTXOPRM macro to reassign new trace IDs.



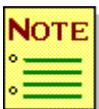
If TRACE1=, tracing is turned off and the trace facility is not active.

If CICS tracing has been disabled, then connector tracing will be disabled.

ABEND operator control command

Issue the ABEND operator control command to abnormally terminate the connection with the PDM, deactivate the CICS Connector, and seal the CICS Connector to prevent further activity. Sealing the Connector flags it as nonusable (see note below). CICS remains fully functional. All PDML commands being processed, as well as future PDML commands, receive a NOTO status.

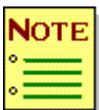
Before the CICS Connector terminates, a dump of all virtual storage areas associated with the CICS Connector is placed on the CICS Dump Data Set. These memory areas appear as a SEGMENT DUMP with a 0998 dump code. Refer to IBM CICS documentation for information on how to print dumps in the CICS Dump Data Set.



Use the ABEND command only in extreme circumstances. Try the DISCONNECT command first. If you issue the ABEND command, you will not be able to use the CICS Connector until you bring CICS down, restart it, and reconnect CICS with the PDM. (The purpose of using the ABEND command is to prevent further use of the CICS Connector to avoid integrity problems.)

DUMP operator control command

Issue the DUMP operator control command to dump all virtual storage areas associated with the CICS Connector. The virtual storage areas appear as a SEGMENT DUMP on the CICS Dump Data Set with a D998 dump code. The DUMP command does not stop the PDM or the CICS Connector from functioning. You may issue it at any time while the CICS Connector is connected. Refer to IBM CICS documentation for information on how to print dumps in the CICS Dump Data Set.



Warning: Do not use codes 0998 and D998 in your programming environment. These codes are reserved for Cincom use only.

System recovery

CICS abends

Should the CICS system abend, restart it using the recovery procedures defined by your systems programmer. If the PDM detects a CICS system abend, it automatically recovers PDM resources.

After restarting CICS, use the `CONNECT` command to reconnect the CICS Connector to the PDM unless the default is set to automatic connect.

PDM abends

Should the PDM abend while the CICS Connector is connected to it, the Connector will detect the abend. All pending PDML commands as well as all PDML commands issued after the PDM failure will receive a `NOTO` status.

Begin recovery by issuing the `DISCONNECT` command to disconnect the Connector from the PDM. If the PDM is running under the attached mode, issue the `CONNECT` command to restart the PDM and reconnect the Connector. If the PDM is running in the central operating mode, restart the PDM and then issue the `CONNECT` command.

If task logging is turned on and you have successfully restarted the PDM, the PDM will recover all PDM resources for all tasks signed on at the time of failure. Tasks signed on but not issuing a `COMMIT` will be signed off.

Concurrent CICS and PDM abends

If CICS and the PDM abend concurrently, restart CICS using the recovery procedures defined by your systems programmer. Restart the PDM using the instructions for recovering PDM system abend.

If the CICS systems programmer specified “automatic connect” as the default at CICS initialization, the CICS Connector is reconnected and you do nothing else. Otherwise, issue the `CONNECT` command to reconnect the Connector to the PDM.

4

Installing the CICS Connector

To install the CICS Connector, you must have an operational CICS system and follow the installation procedures contained in this chapter. When installing these components in an existing CICS system, you must include the library containing any SUPRA Server programs to be loaded by CICS services in the library concatenation list (DFHRPL for OS/390, LIBDEF phase for VSE/AF).

Generate the CICS nucleus to include these facilities:

- ◆ Execute Interface Program (EIP)
- ◆ File Control Program (FCP) to support VSAM KSDS files (if MANTIS is installed), to access VSAM files through RDM.
- ◆ Ensure the DFHSIT macro specifies DFHTSP** for the Temporary Storage Control Program (TSP). ** is the suffix for a temporary storage program which supports use of auxiliary storage. The system generates this module during a CICS installation to support auxiliary storage.
- ◆ Dump Control Program (DCP)
- ◆ Basic Mapping Support (BMS) program if Directory Maintenance or the Connector Information Transaction (ISRA) is installed. PAGING=YES is required.

Refer to the appropriate IBM documentation for information on building a CICS system containing these options. Tailor all other CICS options to your specific requirements.

Defining connector resources to CICS

When you install SUPRA, CICS needs information about SUPRA Server resources. This section describes the resources you must define and the entries you need to make for each. Only required parameters are given. You will need to tailor the optional parameters to your particular environment as needed.

For additional information on defining CICS resources, refer to the appropriate IBM documentation. If you are using CONTROL, MANTIS, RDM, or SPECTRA, refer to the appropriate manual for CICS resource definition.

You must define the following resources:

Resource	Section
Transient Data Destination	“ Transient data destination ” on page 63
Programs and Transactions	“ Programs and Transactions ” on page 67
Program List Table	“ Program List Table (PLT) ” on page 68
System Initialization Table	“ System Initialization Table (SIT) ” on page 70
Transaction List Table	“ Transaction List Table (XLT) ” on page 70

The SUPRA Server installation tape contains sample offline RDO transactions. You can use these members to add or upgrade SUPRA Server resource definitions. For more information, refer to the *SUPRA Server Installation Primer* that accompanies your installation tape.

The following samples are provided in the OS/390 CICSTBLS data set and as .TABLE members in the VSE library:

DFHDCT	(all CICS releases)
DFHPLTCI	(all CICS releases)
DFHPLTDT	(all CICS releases)
DFHSIT	(all CICS releases)
DFHXL	(all CICS releases)
RDOOPER	(CICS connector entries)
RDODIRM	(Directory Maintenance entries)
RDORDM	(Relational Data Manager)
RDOSPEC	(Spectra)
RDOISRX	(Connector PDM Information System)
RDOGNIT	(Security Time Out (RACF-exit to drive RSSO at timeout))
RDOONEW	(Upgrade member for New Connector Program (CSTXRSSO))

Transient data destination

CICS uses the Destination Control Table (DCT) to obtain information for use by the transient data program (data that is transient to CICS). SUPRA Server requires that the destination for the CICS Connector Activity Audit Trail be defined in the DCT. You can route the CICS Connector Activity Audit Trail indirectly to any destination you now use to accumulate hard-copy messages, such as the CSSL destination. You can also route it directly to a hard-copy printer.

OS/390

For sample input to define a transient data destination for the CICS Connector Activity Audit Trail, refer to the DFHDCT member on the OS/390 CICSTBLS library. This entry routes the CICS Connector Activity Audit Trail to DESTID=CSSL.

VSE

For sample input to define a transient data destination for the CICS Connector Activity Audit Trail, refer to the DFHDCT.TABLE member on the VSE library. This entry routes the CICS Connector Activity Audit Trail to DESTID=CSSL.

The following considerations apply to the DCT:

- ◆ Always enable the destination with an OPEN in the DCT. If you define the destination as an indirect destination, also enable its final target. For more information, refer to the appropriate IBM documentation.
- ◆ The destination requires a logical record length of at least 120.
- ◆ If you use a DFHDCT TYPE=INTRA macro in defining the destination for the CICS activity audit trail, you cannot specify DESTRCV=LG.

The following is an OS/390 example of the SUPRA Server indirect entry used within a DCT. (This is an example of a “customized” entry. The sample input provided is not this detailed.)

```

MSGUSR  DFHDCT TYPE=SDSCI,   CICS SHUTDOWN STATS  X
        BLKSIZE=136,                X
        BUFNO=1,                    X
        DSCNAME=MSGUSR,             X
        RECFORM=VARUNB,             X
        RECSIZE=132,                X
        TYPEFLE=OUTPUT

MUSR    DFHDCT TYPE=EXTRA,                X
        DESTID=MUSR,                    X
        DSCNAME=MSGUSR

CSSL    DFHDCT TYPE=EXTRA, STATS          X
        DESTID=CSSL,                   X
        DSCNAME=MSGUSR

MUSR    DFHDCT TYPE=INDIRECT, CICS CONNECTOR LOG  X
        DESTID=MUSR,                   X
        INDEST=CSSL,
        END
    
```

For OS/390, the above example requires the following JCL for a hard-copy printout:

```
//MSGUSR DD SYSOUT=*
```

The following is a VSE example of the SUPRA Server indirect entry used within a DCT:

```

MSGUSR  DFHDCT TYPE=SDSCI,      CICS SHUTDOWN STATS  X
        DEVADDR=SYSLSTSYS001,      X
        BLKSIZE=136,                X
        BUFNO=1,                    X
        DSCNAME=MSGUSR,             X
        RECFORM=VARUNB,             X
        RECSIZE=132,                X
        TYPEFLE=OUTPUT,             X
        DEVICE=1403

CSSL  DFHDCT TYPE=EXTRA, STATS      X
        DESTID=CSSL,                X
        DSCNAME=MSGUSR

MUSR  DFHDCT TYPE=EXTRAINDIRECT,  CICS CONNECTOR LOG  X
        DESTID=MUSR,                X
        DSCNAMEINDEST=MSUUSRCSSL
        END

```

-printer:

```

MUSR  DFHDCT TYPE=EXTRA,          X
        DESTID=MUSR,              X
        DSCNAME=CSIAUDIT

CSIAUDIT DFHDCT TYPE=SDSCI,      X
        DSCNAME=CSIAUDIT,        X
        BLKSIZE=2400,            X
        RECFORM=FIXBLK,          X
        RECSIZE=120,             X
        TYPEFLE=OUTPUT

```

-printout:

```

//CSIAUDIT DD SYSOUT=*

MUSR  DFHDCT TYPE=EXTRA,          X
        DESTID=MUSR,              X
        DSCNAME=CSIAUDIT

```

VSAM files

The CICS connector or Directory Maintenance do not require any VSAM files.

- ◆ The MANTIS Cluster file
- ◆ The MANTIS Transfer file
- ◆ SPECTRA sort files
- ◆ SPECTRA user files
- ◆ SPECTRA RRDS personal files (VSE only)
- ◆ RDM user files

MANTIS files

If you are installing MANTIS files, refer to *AD/Advantage MANTIS Administration OS/390, VSE/ESA, P39-5005*, for the required entries.

SPECTRA files

If you are installing SPECTRA, refer to the *SPECTRA Administrator's Guide, P26-9220*, for details on defining these resources.

RDM files

If you are installing RDM, refer to the *SUPRA Server PDM RDM VSAM Support Supplement (OS/390 & VSE), P26-8222*.

Programs and Transactions

OS/390

To define transactions to CICS, refer to the RDOOPER, RDODIRM, RDORDM, RDOSPEC, RDOONEW, and RDOISRX members in the OS/390 CICSTBLS library.

VSE

To define transactions to CICS, refer to the RDOOPER.TABLE, RDODIRM.TABLE, RDORDM.TABLE, RDOSPEC.TABLE, RDOONEW.TABLE and RDOISRX.TABLE members in the VSE library.



The RDOISRX entries are new with Release 2105 of the Connector. RDOISRX makes the Connector PDM Information System operational. The RDOONEW contains all entries needed to upgrade from Connector release 2103 and up.

SPECTRA

For additional SPECTRA information, refer to the *SPECTRA Administrator's Guide*, P26-9220.

MANTIS

For details on required MANTIS transaction definitions, refer to *AD/Advantage MANTIS Administration OS/390, VSE/ESA*, P39-5005.

RDM

For additional RDM information, refer to the *SUPRA Server PDM RDM Administration Guide (OS/390 & VSE)*, P26-8220.

Program List Table (PLT)

CICS allows you to supply Program List Tables (PLT) containing the names of programs to be executed by CICS at initialization and/or termination. You may supply one PLT for initialization and one for termination.

Initialization PLT



Instead of using the initialization PLT to connect to the PDM, we recommend you use the CICS sequential terminal facility to connect to the PDM, and start any transaction that need to be done at CICS startup.



If you are using CICS/ESA, place the following entry before the other Cincom entries documented below:

```
DFHPLT TYPE=ENTRY,  
PROGRAM=DFHDELIM
```

To automatically connect the CICS Connector to the PDM (optional):

OS/390

Refer to the DFHPLTCI module in the OS/390 CICSTBLS library for the specific entry required.

VSE

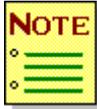
Refer to the DFHPLTCI.TABLE member on the VSE library for the specific entry required.

The following consideration applies to the initialization PLT:

- ◆ You must define PPT entries for the corresponding programs defined in the initialization PLT.

Termination PLT

In the termination PLT, you can supply an entry to automatically disconnect the CICS Connector from the PDM.



Cincom recommends enabling the automatic disconnect feature even if you expect the interface to already be disconnected when CICS terminates. If the interface is already disconnected, the automatic disconnect request is ignored.

For the automatic disconnect of the CICS Connector from the PDM:

OS/390

Refer to the DFHPLTDT member in the OS/390 CICSTBLS library.

VSE

Refer to the DFHPLTDT.TABLE member on the VSE library.

The following considerations apply to the termination PLT:

- ◆ The termination table entry above must precede the CICS required entry for DFHDELIM.
- ◆ If you define a termination PLT, you must define a PPT entry for the corresponding program.

System Initialization Table (SIT)

CICS uses the System Initialization Table (SIT) to supply unique environment information to the CICS system initialization program at CICS startup. The SIT parameters may be assembled into a SIT table and/or supplied at CICS execution time. Supplying the parameters at CICS execution time allows dynamic configuration changes and eliminates the need for table reassembly. Refer to the DFHSIT member in the CICSTBLS library for the required entries.

- ◆ PLTPI=xx, is the suffix from the Program Load Table for the CICS initialization process. The SUPRA distributed SIT table uses a default suffix of CI.
- ◆ PLTSD=xx, is the suffix from the Program Load Table for the CICS shutdown or termination process. The SUPRA distributed SIT table uses a default suffix of DT.
- ◆ XLT=xx, is the transaction suffix from the Transaction List Tables.

Communications SIT parameters:

IRCSTRT=NO CICS/MRO or DL/I inter-region communication startup

ISC=NO CICS/ISC environment inactive

General SIT parameters:

START=COLD CICS/VS system initialization

Transaction List Table (XLT)

CICS allows you to supply a Transaction List Table (XLT) containing the transaction IDs of programs that may be executed during the first stage of CICS shutdown. You should include the Operator Control Program ID in this list so you can communicate with the CICS connector during CICS termination.

For the required parameters, refer to the DFHXLT member of the OS/390 CICSTBLS library or the DFHXLT.TABLE member in the VSE library.

Modifying your CICS execution JCL

Modify your CICS execution JCL to include the DD (DLBL) statements for the various components of SUPRA Server. You may omit the DD (DLBL) statements for SUPRA Server components you are not installing or not executing under a specific copy of CICS.

For the required JCL statements needed for the SUPRA components, refer to the *SUPRA Server PDM and Directory Administration Guide (OS/390 & VSE)*, P26-2250.

OS/390

For OS/390 sample CICS execution JCL, refer to the JCLLIB distributed on the installation tape.

VSE

For VSE, refer to sample JCL members on the VSE library. TXJ\$INDX.A is an index to the distributed members.

For additional CICS execution JCL information, refer to *AD/Advantage MANTIS Administration OS/390, VSE/ESA*, P39-5005, and the *SUPRA Server PDM RDM VSAM Support Supplement (OS/390 & VSE)*, P26-8222.

Generating the operator control parameter table (CSTXOTBL)

The Operator Control Parameter Table (CSTXOTBL) performs the following functions:

- ◆ Supplies defaults for the CONNECT operator control command parameters.
- ◆ Defines the groups of files which may be referred to as groups through the OPEN and CLOSE operator control commands. This function is optional.

Generate an Operator Control Parameter Table by assembling a source module containing calls to the CSTXOPRM macro followed by an assembler END statement. Then, linkedit the resulting object code under the name CSTXOTBL. A default version of this module is supplied in the SUPRA MACLIB and LINKLIB libraries under the name CSTXOTBL. If you need to change CSTXOTBL, disconnect the CICS Connector; create, assemble, and link the new CSTXOTBL; issue the CICS CEMT NEWCOPY command; and reconnect the CICS Connector. For a description of the CSTXOPRM macro, See [“Using Cincom CICS Connector CSTXOPRM macro”](#) on page 135.

The first expansion of the CSTXOPRM macro must supply the information found in the 2 examples listed below. You can use subsequent expansions to define the file groups as shown in example 2 below.

Example 1. This example generates default entries for THREADS, TASKS, MEMORY, and retry counts for TFUL, CFUL and ICOR statuses. It defines the operator control task transaction ID as OPER. The defaults take effect at an automatic connect or, if you do not supply any parameters, at a manual connect.

Parameter values in this example may not meet your actual installation requirements.

```
CSTXOPRM TABLE, THREADS=2, TASKS=2, MEME=9,
CCPTRID=OPER, USERTID=RDMI, MESSQID=MUSR,
TFUL=50, CFUL=50, ICOR=20
END
```

Example 2. This example generates default entries for THREADS, TASKS, and MEME for TFUL, CFUL and ICOR statuses. It defines the operator control task transaction ID as OPER. Two macros have the operand GROUP. The first supplies a group name of TESTREAD for the file definitions following it. The second supplies a group name of TESTWRIT for the file definitions following it:

```
CSTXOPRM TABLE, THREADS=2, TASKS=2, MEME=9,
OCPTRID=OPER, USERTID=RDMI, MESSQID=MUSR,
TFUL=50, CFUL=50, ICOR=20
CSTXOPRM GROUP, GROUPNM=TESTREAD
CSTXOPRM FILE, FILENM=AAAA, OPNMODE=READ, CLSMODE=PART
CSTXOPRM FILE, FILENM=BBBB, OPNMODE=READ, CLSMODE=PART
CSTOXPRM FILE, FILENM=CCCC, OPNMODE=READ, CLSMODE=PART
CSTXOPRM FILE, FILENM=DDDD, OPNMODE=READ, CLSMODE=PART
CSTXOPRM GROUP, GROUPNM=TESTWRIT
CSTXOPRM FILE, FILENM=CCCC, OPNMODE=SUPD, CLSMODE=COMP
CSTXOPRM FILE, FILENM=DDDD, OPNMODE=SUPD, CLSMODE=COMP
END
```

Tailoring RDM options

You can tailor RDM CICS options by coding the C\$VOOPTM macro. For information on coding requirements, refer to the *SUPRA Server PDM RDM Administration Guide (OS/390 & VSE)*, P26-8220. If you do not want to use the Sequential BSAM Terminal feature to start RDM, you can initialize it during the PDM connect. First perform the following steps:

1. Specify the USERTID parameter of the CSTXOPRM macro with the value of a transaction ID (USERID=RDMI).
2. Reassemble and link the CSTXOTBL table.
3. Add the transaction ID RDMI to your CICS tables. RDMI must invoke a program (STARTRDM) that will start RDM.
4. Create the CICS Command Level program to start RDM (STARTRDM) containing the following command:

```
EXEC CICS START TRANSID('OPER') FROM(OPERPARM) LENGTH(11)
```

where OPERPARM is reference to a data area containing the following character string:

```
␣RDM␣START␣
```



Note the leading, embedded, and trailing blanks (␣).

5. Include this program in your CICS application program library.

If PDM Connect processing completes successfully, the CICS Connector will initiate the transaction specified in the USERTID parameter. That transaction will invoke the STARTRDM program, which will start the OPER RDM START transaction. This method ensures that RDM will not be started until the PDM has been connected, but before CICS initialization is complete.

CICS Steplib requirements

If you are running the PDM central the following modules need to be moved from your SUPRA LINKLIB into a library in the CICS steplib concatenation:

- ◆ CSTX050C
- ◆ CSXSDMOD

If you are running CICS with an attached or attached central PDM and need CICS authorized for the performance option, your SUPRA LINKLIB will need to be included into the CICS steplib concatenation and therefore must be authorized. You may follow these steps to authorize this library:

1. Maintain your version of SUPRA TISXA.LINKLIB for CINCOM maintenance.
2. Create a job to move authorized versions of the needed modules from the above SUPRA TIS/XA library into your standard CICS authorized library, per IBM installation requirements.

Tailoring the CICS Connector node error program

In a CICS pseudoconversational environment, you must have a facility to back out PDM tasks associated with terminals that have communication errors. A sample CICS Node Error Program (NEP) is provided (DFHZNEP) which could be assembled and linked into a CICS system immediately to provide this facility.



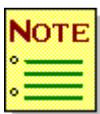
Cincom strongly recommends, however, that you first become familiar with the concept of node error programs by referring to the *IBM CICS Customization Guide*, as node error programs must be designed very carefully. Any third-party software you may have (VTAM session managers) can also have a bearing on the requirements of a node error program.

Tailoring NEP for SUPRA

The CICS Connector Node Error Program (NEP) schedules the LST1 transaction (program CSTXLST1) to recover context for tasks with a failed terminal (see [“How the CICS Connector handles task recovery”](#) on page 28). In an MRO system, the LST1 transaction uses a control table, which is defined by the CSTXMROT macro. You can tailor this macro to your environment by changing the members in the MACLIB below, using the following guidelines:

1. Review DFHZNEP if you already have an active DFHZNEP, determine which changes you need to apply to your existing DFHZNEP).
2. If you *do not* need CICS Connector NEP support across an MRO environment, use the member CSTXLST1 provided on the SUPRA LINKLIB. The source to CSTXLST1 is provided on the SUPRA MACLIB. If you determine that changes need to be made to this member, process CSTXLST1 with the CICS Command Level preprocessor and link edit the resulting member using the linkage editor parameter REUS. Use the CSTXLS1L linkdeck provided on the SUPRA MACLIB to link edit the composite module CSTXLST1. Be sure to specify the linkage editor parameter REUS.

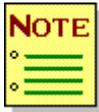
3. If you *do* need MRO support for the CICS Connector NEP, perform the following steps:
 - a. Modify the CSTXMROT source member provided on the SUPRA MACLIB to accurately reflect your MRO transactions.
 - b. Assemble CSTXMROT and linkedit the resulting member using the linkage editor parameter REUS.
 - c. Modify the CSTXLS1L linkdeck member on the SUPRA MACLIB to provide an INCLUDE statement for CSTXMROT. Place the INCLUDE statement for CSTXMROT after the INCLUDE statement for CSTXLST1.
 - d. Linkedit the composite module CSTXLST1 using the modified CSTXLS1L linkdeck. Be sure to specify the linkage editor parameter REUS.



You must provide the following:

- ◆ The remote transaction IDs you specified in your RDO or CICS table definitions.
 - ◆ The remote (AOR) regions with the LST1 programs you specified in the CSTXMROT module.
 - ◆ Do not alter the AMODE/RMODE statements in the link decks. They are set to allow the CICS Connector to be resident in extended storage.
-

4. If you wish to use the CSTXMRXT user exit, perform the following steps:
 - a. Review the sample CSTXMRXT source member provided on the SUPRA MACLIB and make any changes you need for your environment.
 - b. Assemble CSTXMRXT and linkedit the resulting member using the linkage editor parameter REUS.
 - c. Modify the CSTXLS1L linkdeck member on the SUPRA MACLIB to provide an INCLUDE statement for CSTXMRXT. Place the INCLUDE statement for CSTXMRXT after the INCLUDE statement for CSTXLST1 (or after the INCLUDE statement for CSTXMROT, if you are providing CSTXMROT as in Step 3).
 - d. Linkedit the composite module CSTXLST1 using the modified CSTXLS1L linkdeck. Be sure to specify the linkage editor parameter REUS.



You must provide the following:

- ◆ The remote transaction IDs you specified in your RDO or tables.
 - ◆ The remote (AOR) regions with the OPER transaction codes you specified in the MROT table.
 - ◆ If you wish to use the CSTXMRXT user exit, review the sample, make any changes you need to make, and then add the revised sample to the LS1L link control member.
-

Example MRO environment

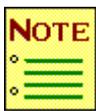
In an MRO environment, there are three possible regions:

- ◆ AOR (Application Owning Region)
- ◆ TOR (Terminal Owning Region)
- ◆ TOR/AOR (Terminal Owning Region that also has some applications running under it)

Terminals are owned by the TOR, and that is the only region that is notified of a VTAM problem. Consequently, the Node Error Program (NEP) will run only in the TOR that owns the terminal that has the error. The NEP also allows only limited functionality.

Because SUPRA allows a pseudoconversational hiatus (no active CICS task for the PDM task) to be signed on in an AOR, you must provide a way to notify the AOR that a terminal has a problem which you need to clear up.

You can do this by adding code to an existing Node Error Program or by using the sample supplied on the SUPRA OS/390 MACLIB or VSE library.



CICS looks for DFHZNEP as the name of the Node Error Program in the PPT. You cannot give it a different name.

The NEP will start the LST1 transaction, passing to it the terminal ID of the terminal with the problem. LST1 (CSTXLST1 module) is responsible for clearing up the problem.

To accomplish this, LST1 issues CICS START commands for the various transaction IDs associated with the CSTXMROT table. (The default, no CSTXMROT table, is that CSTXLST1 issues one START for OPER and OPER then runs wherever it is routed. If multiple AORs are connected to the PDM, CSTXLST1 issues START commands for all transaction IDs in the CSTXMROT table.)

CSTXMROT is simply an assembly of a table that can be included in the composite linkedit for CSTXLST1. The CSTXMROT module, in order to conform to the layout required for the table manipulation, is simply an expansion of the C\$TXMROT macro, which is in the OS/390 MACLIB and the VSE library. The parameters of the C\$TXMROT macro are as follows:

- ◆ TRANSID= An appropriate transaction ID, which must be defined to CICS.
- ◆ COMMENT= A field that you can use to track what the transaction ID is for (COMMENT= 'CICS abc'). The maximum length of this field is 11 bytes.
- ◆ MESSQID= The first entry. This is the message queue ID CSTXLST1 uses to route its messages (the default is MUSR).

You need as many C\$TXMROT entries as there are AORs. If you use CSTXMROT, you will need to start a transaction to handle the TOR connections to the database if connections are allowed from the TOR.

The following is an example MRO environment configuration:

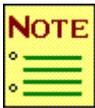
CICS A	CICS B	CICS C	CICS D
TOR/AOR	AOR	AOR	TOR
connected to PDM	connected to PDM	connected to PDM	no PDM connection
PCT LST1 (local)	no LST1	no LST1	LST1 (local)
OPER (local)	no OPER	no OPER	OPER (remote)
OPRB (remote)	OPRB (local)	no OPRB	OPRB (remote)
OPRC (remote)	no OPRC	OPRC (local)	no OPRC

In this configuration, CICS A communicates with CICS B and C, and CICS D communicates with CICS A and B (but not C). You would need the following CSTXMROT table for CICS A:

```
C$TXMROT TRANSID=OPER,COMMENT='CICS A',MESSQID=aaaa
C$TXMROT TRANSID=OPRB,COMMENT='CICS B'
C$TXMROT TRANSID=OPRC,COMMENT='CICS B'
```

You would then need the following CSTXMROT table for CICS D:

```
C$TXMROT TRANSID=OPER,COMMENT='CICS A',MESSQID=aaaa
C$TXMROT TRANSID=OPRB,COMMENT='CICS B'
```



In this example you would need 2 different CSTXLST1 composite modules (different CSTXMROT tables), or you could run the same one and ignore a TRANSID error for CICS Ds START of OPRC.

The CSTXMRXT exit point

The CSTXMRXT exit point allows you to change the TERM-ID if necessary. Although some users change TERM-IDs when going from one region to another, there is no mechanism to do this automatically (you must code this exit in your environment.)



Caution: Changing the TERM-ID is not recommended.

You can also specify that you do not want a particular transaction started. To do this, you need to add 2 input parameters:

1. The first parameter is the address of the field that contains the TERM-ID.
2. The second parameter is the address of the CSTXMROT table. The table entry length is 16 bytes and is terminated by an x'FF' in at least the first character of the TASK-ID. The format of this table is as follows:
 - a. A 4-byte TASK-ID.
 - b. A 1-byte FLAG (The default setting is Y; if you set the flag to N, the EXEC CICS START for the appropriate TASK-ID is not performed.)
 - c. An 11-byte comment (defined by COMMENT=).

5

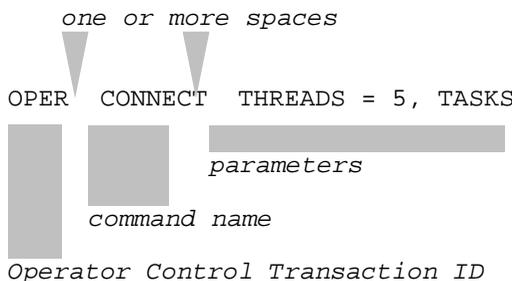
Using operator control commands

Operator control commands (OPER commands) have English names describing the specific operation they invoke (CLOSE, CONNECT, DISCONNECT, DUMP, OPEN, STATUS, PURGE). To enter these commands, you can use any terminal under CICS control including a CICS system console. A higher CICS security level should be assigned to the OPER transaction to prevent unauthorized use. You can enter the full name of the command or abbreviate it to a minimum of 3 characters. For example: CLOSE, CLOS and CLO are equivalent.

The operator control commands (OPER commands) return messages to the terminal that issued the command and to the message queue specified in the CICS Connector Operator Control Parameter Table Macro (CSTXOPRM). See [“Using Cincom CICS Connector CSTXOPRM macro”](#) on page 135 for information on the CSTXOPRM macro.

To issue any operator control command, you need an Operator Control Transaction ID. You define this 4-character ID as a transaction to CICS in the OPCTRID parameter of the Operator Control Parameter Table (CSTXOTBL). See [“Generating the operator control parameter table \(CSTXOTBL\)”](#) on page 72 for more information on using CSTXOTBL.

To issue a command, first clear the screen. Then enter your Operator Control Transaction ID, one or more spaces, the command name, one or more spaces, and the parameters, if any. You can enter keyword parameters in any order. You may use commas or spaces between parameters. Do not use commas between the command name and the first keyword. The following figure shows the correct command entry format:



ABEND command

ABEND stops the CICS connector from functioning. Use it only if an interface malfunction jeopardizes the integrity of the database.



Cincom recommends that you issue a DISCONNECT command first to disconnect the interface from the PDM.

If you issue an ABEND, you must also bring CICS down and restart it to reconnect with the PDM. The CICS Connector cannot operate properly again until you do this.

ABEND

Description *Required.* Stops the interface from functioning.

Consideration An abend can occur only if the CICS Connector is connected.

CLOSE command

Use CLOSE to close individual database files or a group of files. The Operator Control Parameter Table (CSTXOTBL) contains file groups and their close modes (see “[Tailoring RDM options](#)” on page 74). This command closes files only if the OPENX-OPTION=PROCESS is specified in the active environment description. (Contact your DBA for this information.)

$$\text{CLOSE} \left\{ \begin{array}{l} \text{FILE} = \text{ffff} \left[\text{MODE} = \left\{ \begin{array}{l} \text{COMP} \\ \text{PART} \end{array} \right\} \right] \\ \text{GRO[UP]} = \text{gggggggg} \end{array} \right\}$$

$$\left\{ \begin{array}{l} \text{FILE} = \text{ffff} \\ \text{GRO[UP]} = \text{gggggggg} \end{array} \right\}$$

Description	<i>Required.</i> Specifies a database file or a group of files to be closed.
Format	<i>ffff</i> 4 alphanumeric character file name
	<i>gggggggg</i> 1–8 alphanumeric character group name
Options	FILE Specifies a single database file.
	GROUP Specifies a group of files.

Considerations

- ◆ The CICS Connector must be connected in order to close a file.
- ◆ If an error occurs while closing any file in a group, all files remain in their original states. The name of the file receiving the error is displayed with a message. The CICS Connector Activity Audit Trail lists all Close commands and resulting messages (see “[Opening and closing database files dynamically](#)” on page 50).

MODE = $\left\{ \begin{array}{l} \text{COMP} \\ \text{PART} \end{array} \right\}$

Restriction	Use only with the FILE parameter.	
Description	<i>Optional.</i> Specifies the mode to be used to close the file.	
Default	COMP	
Options	COMP	Use for an operating system close. COMP unlocks the file, if necessary, and resets the file mode to null. A task may access the file only to open it.
	PART	Use for a non-operating system close. PART unlocks the file, if necessary, and resets the file mode to IUPD. Any task may read the file, but no task may update it.

CONNECT command

Use CONNECT to connect the CICS Connector to the PDM. If you set up the CICS Connector for automatic connection at CICS initialization, CONNECT is issued with default values. See the Operator Control Parameter Table (CSTXOTBL) in “[Generating the operator control parameter table \(CSTXOTBL\)](#)” on page 72 for the default values.

If you use the CICS sequential terminal options to connect at CICS startup, you may specify the parameters listed below. Any parameter not specified will use the default parameters from CSTXOTBL.

The system rejects a request to connect an interface that is already connected. To see if the CICS Connector is connected, issue a STATUS command.

```
CONNECT [THREADS=nnnnn] [,TASKS=nnnnn] [,MEME=nn]
        [,TFUL=nnnnn] [,CFUL=nnnnn][,ICOR=nnnnn]
```

THREADS=nnnnn

- | | |
|----------------------|---|
| Description | <i>Optional.</i> Specifies the number of threads for communication between the CICS Connector and the PDM. |
| Default | The value of the THREADS parameter of CSTXOTBL |
| Format | 1–32767 |
| Consideration | To tune this parameter, refer to the SUPRA Server PDM Tuning Guide (OS/390 & VSE) , P26-0225. |

TASKS=nnnnn

Description *Optional.* Specifies the maximum number of CICS tasks that may be signed on to the PDM at any one time.

Default The value of the **TASKS parameter** of CSTXOTBL

Format 1–32767

Considerations

- ◆ Refer to the *SUPRA Server PDM Tuning Guide (OS/390 & VSE)*, P26-0225, for information on tuning this parameter.
- ◆ The CICS Connector reserves one task in addition to the number of tasks you specify in the OPER CONNECT request or in the CSTXOTBL module. This extra task is for Disconnect processing. You must ensure that the REALM environment description used by the PDM has allowed an additional task for each CICS system that will use this PDM. For more information on the requirements for threads, tasks, and interfaces in the REALM environment description, refer to the *SUPRA Server PDM and Directory Administration Guide (OS/390 & VSE)*, P26-2250.

MEME=nn

Description *Optional.* Specifies the maximum number of 64K memory extensions the connector will automatically attempt before returning the ICOR status because of lack of CICS Connector memory.

Default The value of the **MEME parameter** of CSTXOTBL

Format 0–99

Considerations

- ◆ You can still receive ICORS from the PDM. This parameter concerns only the memory for the CICS Connector.
- ◆ The CICS Connector calculates expected memory usage at connect time. This parameter allows for automatic extension to that allocation if necessary. The OPER STAT command displays the memory usage and should be monitored.

TFUL=nnnnn

Description	<i>Optional.</i> Specifies the maximum number of automatic retries if a TFUL status occurs.
Default	The value of the TFUL parameter in the CSTXOTBL
Options	0–32767

Considerations

- ◆ The CICS Connector always tries each PDML command once and then retries as many times as necessary, up to the limit specified in this parameter.
- ◆ TFUL usually means that there are not enough threads available to handle the request.

CFUL=nnnnn

Description	<i>Optional.</i> Specifies the maximum number of automatic retries if a CFUL status occurs.
Default	The value of the CFUL parameter in the CSTXOTBL
Format	0–32767

Considerations

- ◆ The CICS Connector always tries each PDML command once and then retries as many times as necessary, up to the limit specified in this parameter.
- ◆ CFUL usually means that there are not enough tasks available to handle the request.

ICOR=nnnnn

Description *Optional.* Specifies the maximum number of automatic retries if an ICOR status occurs.

Default The value of the **ICOR parameter** in the CSTXOTBL

Format 0–32767

Considerations

- ◆ The CICS Connector always tries each PDML command once and then retries as many times as necessary, up to the limit specified in this parameter.
- ◆ ICOR usually means that there is not enough memory available to handle the request.

DISCONNECT command

Use DISCONNECT to disconnect the CICS Connector from the PDM. When the PDM is in the attached mode, DISCONNECT ends and then detaches the PDM. Any applications using this PDM in central operating mode receive NOTO. See “[Disconnecting the CICS Connector from the PDM](#)” on page 47 for more information on disconnecting the CICS Connector from the PDM.

If you set up automatic disconnection on CICS termination, it is not necessary to issue an explicit DISCONNECT command unless you want to disconnect. The system does not disconnect an interface already disconnected and writes a warning message on the CICS Connector Activity Audit Trail. If you issue the command in such a situation, the warning message also appears on your terminal.

DISCONNECT FORCE = $\left. \begin{array}{l} \text{YES} \\ \text{NO} \\ \text{AUTO} \\ \text{SINF} \end{array} \right\}$

FORCE = $\left. \begin{array}{l} \text{YES} \\ \text{NO} \\ \text{AUTO} \\ \text{SINF} \end{array} \right\}$

Description	<i>Optional.</i> Defines the type of disconnect.	
Default	NO	
Options	<u>Y</u> ES	Disconnects the CICS Connector from the PDM even if CICS tasks are signed on.
	<u>N</u> O	Disconnects the CICS Connector only if no CICS tasks are signed on.
	<u>A</u> UTO	Disconnects the CICS Connector as soon as all currently signed-on CICS tasks issue sign-off.
	<u>S</u> INF	Resets and signs off existing tasks signed on to the PDM.

Considerations

- ◆ Using FORCE=AUTO frees the terminal for other activity.
- ◆ Using FORCE=YES or FORCE=NO displays run-time statistics after a successful disconnection. Statistics are also written on the CICS Connector Activity Audit Trail. See the STATUS command for a list of run-time statistics.
- ◆ If you use FORCE=AUTO and the PDM shows active tasks but the interface shows none, the system changes FORCE=AUTO to FORCE=YES.
- ◆ If FORCE=SINF is unsuccessful, then FORCE=SINF is automatically converted to a FORCE=YES. If FORCE=SINF successfully signs off all tasks (or finds none to sign off), FORCE=SINF is automatically converted to a FORCE=NO.
- ◆ SINF and AUTO also prevent new sign-ons.
- ◆ The FORCE=YES option forces the CICS Connector to free the allocated storage and disable the Task related User Exits (TRUEs). Consequently, if you are in a “stalled” state (if the TRUEs are enabled but connection to the PDM is impossible due to an incomplete connect), issue the OPER DISCONNECT FORCE=YES option to clear up the problem.

You should also use this option if you get a CSTX0791 message and it indicates CLEAN=NO. Issue an OPER DISCONNECT FORCE=YES. The CSTX0791 message will display again, this time indicating CLEAN=YES. At that point, you should be able to reconnect to the PDM.

SUPRA CICS Connector uses a CICS Task Related User Exit to communicate with the MTMT PDM Interface. For more information, refer to the *IBM CICS Customization Guide*.

DUMP command

Use DUMP to place a dump on the CICS Transaction Dump Data set. This dump lists all virtual storage areas used by the CICS Connector. The interface must be connected to the PDM for the DUMP command to execute. DUMP does not interfere with the normal operation of the CICS Connector.

DUMP

Description Writes the contents of the CICS Connector memory areas to the CICS Transaction Dump Data set.

General consideration

OPEN, CLOSE, ABEND and DISCONNECT FORCE=YES processing may internally issue the DUMP command for certain conditions.

OPEN command

Use OPEN to open individual database files or a group of files. The Operator Control Parameter Table (CSTXOTBL) contains file groups and their open modes. This command opens files only if OPENX-OPTION=PROCESS is coded in the active environment description.

$$\text{OPE[N]} \left\{ \begin{array}{l} \text{FILE} = \text{ffff} \left[\text{MOD[E]} = \left\{ \begin{array}{l} \text{IUPD} \\ \text{SUPD} \\ \text{READ} \end{array} \right\} \right] \\ \text{GRO[UP]} = \text{gggggggg} \end{array} \right\}$$

$$\left\{ \begin{array}{l} \text{FILE} = \text{ffff} \\ \text{GRO[UP]} = \text{gggggggg} \end{array} \right\}$$

Description *Required.* Specifies a database file or a group of files to be opened.

Format *ffff* 4 alphanumeric character file name
gggggggg 1–8 alphanumeric character group name

Options FILE Specifies a single database file.
 GROUP Specifies a group of files.

Consideration If an error occurs while opening any file in a group, all files remain in their original states. The name of the file receiving the error is displayed with a message. The CICS Connector Activity Audit Trail lists all OPEN commands and resulting messages.

$$\text{MOD[E]} = \left\{ \begin{array}{l} \text{IUPD} \\ \text{SUPD} \\ \text{READ} \end{array} \right\}$$

Restriction	Use only with the FILE parameter.	
Description	<i>Optional.</i> Specifies the mode to be used to open the file or a group of files.	
Default	SUPD	
Options	IUPD	Marks the file for read only. The file can be read by all tasks but cannot be updated by any task using this PDM. The file will not be locked.
	SUPD	Marks the file for shared updating. The file is locked and can be updated by any task using the same copy of the PDM.
	READ	Marks the file for read only. The file can be read by any task but cannot be updated by any using this PDM. The file is not locked.

General considerations

- ◆ Whenever a primary or related file is opened for SUPD, the related index files are also opened for SUPD. If CLOSE PART is done on primary or related files, the index file remains locked. To unlock the index file, you must issue a CLOSE COMP for all primary and related files with secondary keys stored in the index file.
- ◆ Refer to the [SUPRA Server PDM and Directory Administration Guide \(OS/390 & VSE\)](#), P26-2250, for additional information on the OPEN command.

PURGE command

Use PURGE to purge a SUPRA PDM task from the system. Use this command *very cautiously!* Once you have purged a task, you cannot retrieve it.

PURGE ID = $\left. \begin{array}{l} \mathbf{Txxx} \\ \mathbf{Nnnnnn} \end{array} \right\}$

$\left. \begin{array}{l} \mathbf{Txxx} \\ \mathbf{Nnnnnn} \end{array} \right\}$

Description *Required.* Specifies the ID to be purged or the ID relating to terminal-related tasks.

Format $Txxx$ T, followed by the 4-character CICS terminal ID.
 $Nnnnnn$ N, followed by the 5-character CICS transaction number.

Consideration Under normal circumstances, the CICS Connector NEP recovery routine CSTXLST1 will start a background (nonterminal) OPER task with a “PURGE ID= $Txxx$ ” to clean up the resources associated with the problem terminal.



See “[How the CICS Connector handles task recovery](#)” on page 28 for a description of task cancellation and task recovery.

RDM command

Use RDM to initialize RDM, and to stop and reinitialize RDM without shutting down and restarting CICS.

RDM { START
STOP
FORCE }

Description	<i>Required.</i> Specifies which command to send to RDM for processing.	
Options	<u>START</u>	Initializes RDM, which opens global views (if any) and makes RDM available for processing RDM application requests.
	<u>STOP</u>	Stops RDM, unless there are active (signed-on) RDM applications. RDM storage is freed and RDM modules are released from storage. If RDM applications are still active, you will receive an error reason code.
	<u>FORCE</u>	Stops RDM, even if there are active (signed-on) RDM applications. RDM storage is freed and RDM modules are released from storage. Any RDM applications that were still signed on will receive errors or abends on their next RDML call due to loss of context.

Considerations

- ◆ These commands are useful for bringing new RDM options or maintenance into your system without having to shut down your CICS systems.
- ◆ You can prevent RDM from loading new copies of the RDM modules shown below by disabling them under CICS prior to issuing an RDM STOP or FORCE (using CEMT or CECI). If you disable any of the modules listed below, the RDM STOP or FORCE commands return an error reason code and then continue with the STOP or FORCE processing:

CSVNVRUN	CSVNPLVS	CSVCWORK	CSVOOPTM
CSVCDLI	CSVNVSAM	CSVNDATB	
- ◆ After the RDM STOP or FORCE command completes (and any error reason code displays on the terminal and in the CICS Connector message queue), you can issue an RDM START. After the RDM START completes, you should enable any modules you disabled prior to the RDM STOP or FORCE.

STATUS command

Use STATUS to check if the CICS Connector is connected (active) or disconnected (inactive) to the PDM. If the CICS Connector is connected, use STATUS to look at run-time statistics. Using STATUS does not reset run-time statistics.

STATUS

Description *Required.* Checks whether the CICS Connector is connected.

The following statistics appear on your terminal and on the CICS Connector Activity Audit Trail:

```

CSTX039I : OPER STAT  09:07:10
CSTX053I : THREADS = 00002, TASKS = 00005, TFUL RETRIES = 00050      09:07:10
CSTX053I : CFUL RETRIES = 00050, ICOR RETRIES = 00020, MEME = 00025  09:07:10
CSTX053I : IFRTRY = 00000005, MAXPACK = 00000000, IMMEDRS =        09:07:10
CSTX053I : SYNC = N, TOTC = N, PUSHPOP = N, TRACE1 = 05, TRACE2 = 06 09:07:10
CSTX053I : TRACE3 = 07, MESSQID = MUSR, OCPTRID = OPER, OCPTRTM = NO 09:07:10
CSTX053I : USERTID = NO , TDUE = NO , TPOP = NO                    09:07:10
CSTX053I : TPOS = NO , TPRE = NO , TPRP = NO                      09:07:10
CSTX053I : TSYN = NO , TUID = NO                                   09:07:10
CSTX053I :                                                         09:07:10
CSTX022I : THE SUPRA CICS CONNECTOR IS CURRENTLY CONNECTED.....    09:07:10
CSTX042I : SIGNED ON TASKS = 00000, PEAK SIGNED ON TASKS = 00001 ... 09:07:10
CSTX043I : CFUL STATS: RETRIES = 0000000000, RETURNED = 0000000000 . 09:07:10
CSTX043I : TFUL STATS: RETRIES = 0000000000, RETURNED = 0000000000 . 09:07:10
CSTX043I : ICOR STATS: RETRIES = 0000000000, RETURNED = 0000000000 . 09:07:10
CSTX080I : MEMORY TOTAL = 000000000065504, FREE = 000000000053136 .. 09:07:10
CSTX020I : REQUEST COMPLETED NORMALLY. CLEAR SCREEN TO CONTINUE.... 09:07:10

```

6

Using Cincom CICS Connector exits

Cincom provides user exits for the CICS Connector for the following purposes:

- ◆ TUID Exit provides an exit from the DML Interface processing at the beginning of a CICS task.
- ◆ CSTXUDAT Exit provides an exit from the DATBASC module.
- ◆ TPRP Exit provides an exit from the DML before task processing.
- ◆ TPRES and TPOS Exits provide exit points before and after a PDML is passed to the PDM.
- ◆ TPOP Exit provides an exit from the DML after task processing.
- ◆ TSYN Exit provides synchronization between tasks during processing.
- ◆ TDUE Exit provides processing to interrogate the signed-on table at CICS task detach time and to turn RSSO processing off or on.
- ◆ CSTXMRXT Exit provides a user exit point in the recovery scheduled by the CICS Connector NEP.

All of the exits except CSTXUDAT and CSTXMRXT are loaded during OPER CONNECT. Because all of these exits work together, you should read the section for each one.

Several user exits have been withdrawn and are no longer available under SUPRA CICS/ESA. A brief discussion of these obsolete exits appears in “[Obsolete user exits](#)” on page 128.

Overview of the user exits

All of the user exits for the SUPRA CICS Connector are loaded automatically during OPER CONNECT processing except the USER DATBASC exit, which must be linked with the user application along with DATBASC, and CSTXMRXT, which must be link edited with CSTXLST1. All macros and source code are located in the SUPRA MACLIB. Refer to this library for the definitions of macros, data field names, and source modules.

You define the presence of a particular exit (except for CSTXUDAT and CSTXMRXT) by supplying the name of the program module as the value for one of the exit parameter entries on the CSTXOPRM macro, which defines the CSTXOTBL defaults table. A description of the parameters and how you code them is included in the description of each exit. The parameter areas passed to each exit have a common header, so the exit code can be provided in one exit module which determines internally which exit is being called and then transfers control to that section of code.

The CSTXUSRX module is a working sample user exit module that contains logic for each of the exit points defined. You linkedit this module with the CSTXUSER link deck. The composite module name is CSTXUSER. You can modify CSTXUSRX to perform whatever functions you need at a given exit point. That exit point's parameter in CSTXOPRM will then reference CSTXUSER.

For example, TUID=CSTXUSER coded on CSTXOPRM for the CSTXOTBL defaults table tells OPER CONNECT to load CSTXUSER and place its entry point in the internal reference for the Task User ID Exit (TUID). When CICS Connector processing reaches the Task User ID Exit point (TUID) for a given task, CSTXUSER is called. The common exit type flag is interrogated to identify the exit type, and the logic at USERUTID in CSTXUSRX is executed to manage the TUID exit request.

There is a macro for each of the exit points. For example, C\$XTUID defines the parameters and saved areas passed to the exit for the TUID exit point within the SUPRA CICS Connector.

Internal documentation is supplied in the CSTXUSRX module and in the macros that will help you determine which exits you need and where to place customized exit code.

The SUPRA CICS Connector is based on IBM's Resource Manager Interface concept (RMI) and uses a Task Related User Exit (TRUE) to pass control from the user application to the PDM. Refer to the *IBM CICS Customization Guide* for your particular release of CICS for a description of TRUEs.

The exits described in this chapter are based on exit points from the TRUE logic paths. Refer to the IBM CICS documentation for an explanation of which logic paths are provided. For a visual representation of where the exits fit into the logic flow of your PDML, see the diagrams of the exits on the following pages.

CICS Connector macros

The following are descriptions and usage notes for the CICS Connector macros:

- ◆ **C\$TXCMUE** (Common User Exit PARM and work area). You enter all standard loaded SUPRA CICS user exits with a standard parameter header and a standard work area header, which allows a single program to manage all exit requests. (For more information, refer to CSTXUSRX on your SUPRA Server MACLIB.)
- ◆ **C\$TXMROT** (Define MRO Remote Transactions for CSTXOPER to CSTXLST1). This macro allows you to define the remote transaction IDs that will start the OPER PURGE request in the remote MRO region(s) associated with an out-of-service terminal detected by node error program DFHZNEP. CSTXLST1 manages SUPRA Server resources associated with the out-of-service terminal.
- ◆ **C\$TXTDUE** (Task Detach User Exit PARM and work area). This macro redefines the user exit interface parameter and work area defined by C\$TXCMUE. It provides the specific parameters and work fields associated with the Task Detach User Exit. (For more information, refer to the USERTDUE label in CSTXUSRX on your SUPRA Server MACLIB.)
- ◆ **C\$XTTPOP** (Post Processing User Exit PARM and work area). This macro redefines the user exit interface parameter and work area defined by C\$TXCMUE. It provides the specific parameters and work fields associated with the Post Processing User Exit. (For more information, refer to the USERTPOP label in CSTXUSRX on your SUPRA Server MACLIB.)
- ◆ **C\$XTTPOS** (Post-DML User Exit PARM and work area). This macro redefines the user exit interface parameter and work area defined by C\$TXCMUE. It provides the specific parameters and work fields associated with the Post-DML User Exit. (For more information, refer to the USERTPOS label in CSTXUSRX on your SUPRA Server MACLIB.)

- ◆ **C\$TXTPRE** (Pre-DML User Exit PARM and work area). This macro redefines the user exit interface parameter and work area defined by C\$TXCMUE. It provides the specific parameters and work fields associated with the Pre-DML User Exit. (For more information, refer to the USERTPRE label in CSTXUSRX on your SUPRA Server MACLIB.)
- ◆ **C\$TXTPRP** (Pre-Process User Exit PARM and work area). This macro redefines the user exit interface parameter and work area defined by C\$TXCMUE. It provides the specific parameters and work fields associated with the Pre-Process User Exit. (For more information, refer to the USERTPRP label in CSTXUSRX on your SUPRA Server MACLIB.)
- ◆ **C\$XTSYN** (User Synchronization PARM and work area). This macro redefines the user exit interface parameter and work area defined by C\$TXCMUE. It provides the specific parameters and work fields associated with the User Synchronization Exit. (For more information, refer to the USERTSYN label in CSTXUSRX on your SUPRA Server MACLIB.)
- ◆ **C\$TXUTID** (User Task ID Exit PARM and work area). This macro redefines the user exit interface parameter and work area defined by C\$TXCMUE. It provides the specific parameters and work fields associated with the User Task ID Exit. (For more information, refer to the USERUTID label in CSTXUSRX on your SUPRA Server MACLIB.)

The TUID user exit

The TUID exit is called only at the first CALL DATBAS for the physical CICS task (not transaction). Its primary function is to pass the computed SUPRA task ID for inspection and to allow you to change the SUPRA task ID (the last seven bytes). You can also use this exit to change the Resource Synchronization options for the task.

If a Signed On Table Block (SOTB) is detected as the final SUPRA TASK ID at the time this exit computes or modifies the SUPRA TASK ID (a task was already signed on using the FINAL SUPRA TASK ID), this SOTB information is also passed to UTID.



SOTB returns SUPRA PDM status information. An SOTB is created for each task defined on the OPER CONNECT TASK = parameter.



If you have any functions in your old (Supra135) CSTXUTID exit that were called at each DML or were called from a specific module (from the CSTXEXIT during dynamic backout), you should migrate these functions to the new pre- or post-command exits. (Refer to the *SUPRA Server PDM Migration Guide (OS/390 & VSE)*, P26-0550, for more information.) When you migrate these functions, you must take into the account the new Resource Manager Interface (RMI) architecture and the requirements for CICS/ESA. Refer to the Task Related User Exit documentation in your IBM *CICS Customization Guide* for more information on RMI architecture.

The TUID exit has a more limited function than the old (Supra135) CSTXUTID exit it replaces. Registers 1 and 13 are used on entry to TUID.

You define the TUID exit with the TUID parameter in the CSTXOPRM macro.

- ◆ If you code a user name for TUID in your CSTXOPRM macro, that user name will be the name of the pre-DML exit module loaded during OPER CONNECT.
- ◆ If you do not code a user name in CSTXOPRM, the macro automatically codes the default name of NO, and the exit address will be set to zero during OPER CONNECT.

Register 1 points first to the data area defined by the C\$TXUTID macro. The parameters and options directly related to the required TUID functions are as follows:

- ◆ If the call type flag indicates pass/modify ID (UTIDCTYP = UTIDPMID), these conditions apply:
 - The originally computed task ID is passed to UTIDID.
 - You can modify the last seven bytes of the task ID (UTIDTIDC).
 - You can set the resource synchronization (UTIDSYVF) to any valid value, unless you previously set it to UTIDSYVX, which means NO TLR, NO SYNC. ALLOWED.
 - UTIDSOTF is set to zeroes and has no meaning.
- ◆ If the call type flag indicates passed/signed-on (UTIDCTYP = UTIDPSOT), these conditions apply:
 - A signed-on table entry for the Task ID in UTIDTID has been located.
 - The resource synchronization (UTIDSYVF) contains a copy of the signed-on table's synchronization flag.
 - UTIDSOTF contains a copy of the signed-on table's status flag.

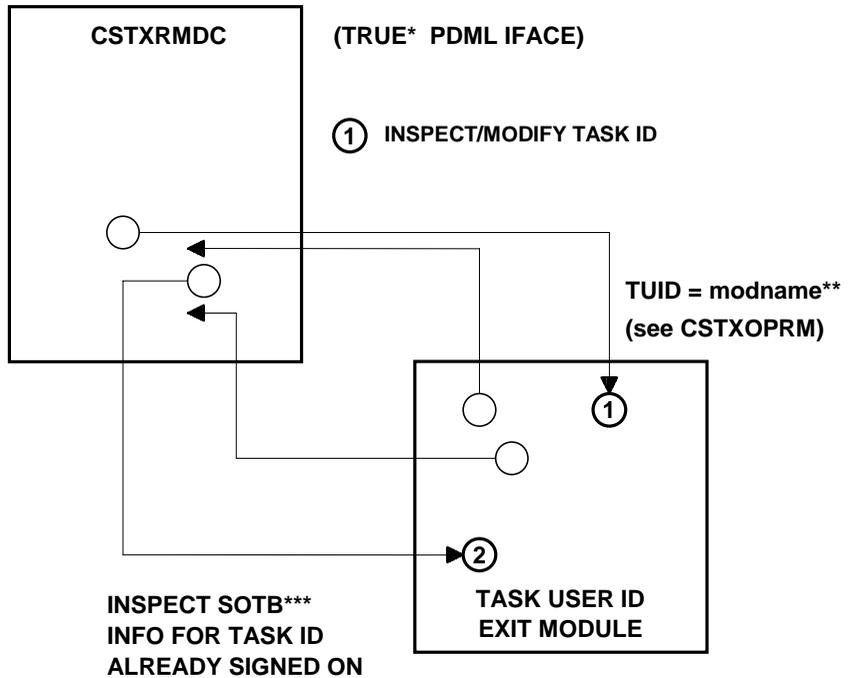
Register 13 points to a save work area with the following layout:

UTIDSAVE DS	18F	SAVE AREA
UTIDCSVE DS	18F	EXTRA SAVE AREA FOR USER CALLS
UTIDWORK DS	18F	USER WORK AREA
UTIDTCOM DS	A	ADDRESS OF 18F USER TASK COMMON AREA
UTIDGADR DS	A	ADDRESS OF 18F USER GLOBAL COMMON AREA

- ◆ This is a task-unique area.
- ◆ UTIDTCOM is used for communication between all user exits for a specific task.
- ◆ UTIDGADR is a global area in the global TRUE work area that is the same for all exits using that TRUE work area. It is used to communicate between user exits across tasks for the life of the current connect.

Because no conditional processing options are provided, the return code value in Register 15 is ignored during return from the user exit.

The following figure illustrates the TUID user exit.



- * Task Related User Exit
- ** See CSTXUSER:USERTID
- *** Signed On Table Block

The CSTXUDAT user exit

CSTXUDAT provides an exit called from the CICS DATBAS module (DATBASC). Source for a sample CSTXUDAT program containing the CSTXUDAT entry is distributed on the SUPRA MACLIB; a linkedited version of this sample is distributed on the SUPRA LINKLIB.

All user exits except CSTXUDAT and CSTXMRXT are loaded automatically at OPER CONNECT time. To use CSTXUDAT, you must link the module containing the CSTXUDAT entry into the composite load module containing DATBASC. For example, to call this exit from a command-level user application program (TESTPGM in the example), you will need a linkdeck as follows:

```

INCLUDE CICSLIB(DFHEAI)
INCLUDE LINKLIB(DATBASC)
INCLUDE LINKLIB(TESTPGM)
INCLUDE LINKLIB(CSTXUDAT) (sample user DATBAS exit with a
                           CSTXUDAT entry)
INCLUDE CICSLIB(DFHEAIO)
ENTRY TESTPGM
AMODE(31), RMODE(ANY)
NAME TESTPGM(R)

```

You can also add your CSTXUDAT exit program to Cincom linkdecks that include DATBASC.

If you code your own CSTXUDAT module, you must supply standard operating system call conventions; specifically, CSTXUDAT is expected to save and restore the general purpose register environment.

When DATBASC receives control from an application program, register saves and base addressing are completed first, and a WEAK EXTERN is created for CSTXUDAT. If the external reference to CSTXUDAT is resolved, control passes to CSTXUDAT.

The following 2 registers pass to CSTXUDAT:

1. Register 1 points to the standard DML parameter address list passed by the caller to DATBAS.
2. Register 13 points to a special save work area with the following layout:

```
USERSAVE DS    18F    SAVE AREA
USERWORK DS    18F    WORK AREA
```

Userwork is a static area within the DATBASC module. Usersave +4 contains the address of the original save area passed by the original caller to DATBASC.

After returning from CSTXUDAT, DATBASC restores register 13 to the original save area. It also restores registers to the original application program's values except for:

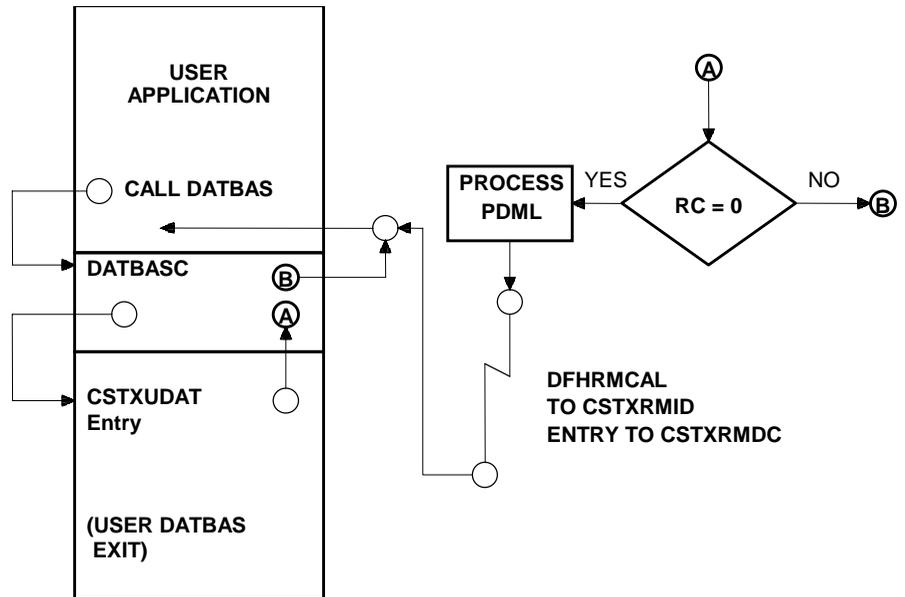
- ◆ Register 12: DATBASC program base register
- ◆ Register 15: Used later by DATBASC to call the PDM interface program

Because DATBASC uses register 15 to call the PDM interface program, it is not available to your application program to be tested for a return code from CSTXUDAT. To alter your application program's processing flow, you must modify the register 14 value saved in the original save area; CSTXUDAT saves the original save area address at USERSAVE+4.



Warning: You must provide task-unique context if you wish to retain information across a DML line.

The following figure illustrates the CSTXUDAT user exit.



The TPRP and TPOP user exits

TPRP and TPOP replace the Modified TOTINT TOTAL Compatibility exit.

The TPRP (preprocess) user exit

You define the TPRP exit with the TPRP parameter in the CSTXOPRM macro.

- ◆ If you code a user name for TPRP in your CSTXOPRM macro, that user name will be the name of the user pre-process exit module loaded during OPER CONNECT.



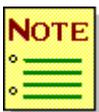
If you code the user name TPRP=CSTXUSER, you can update the TPRP sample user logic in the CSTXUSRX sample source to suit your needs. Use the CSTXUSER linkdeck to install the changed CSTXUSER.

- ◆ If you do not code a user name in CSTXOPRM, the macro automatically codes the default name of NO, and the exit address will be set to zero during OPER CONNECT.

Register 1 points first to the data area defined by the C\$XTPRP macro. (Refer to that macro for a detailed description of all parameters and functions.) The parameters and options directly related to the required TPRP functions are as follows:

- ◆ TPRPDML points to the current PDML parameter address list.
- ◆ The TPRPDMLF DML flag determines whether or not the physical DML will be processed. You can set this flag (yes or no).
- ◆ The TPRPTOTF TOTAL COMPATIBILITY flag determines whether or not compatibility with TOTAL is required. You can set this flag (REQUIRED or NOT REQUIRED).

If you set this flag to REQUIRED, the macro calls the CSTXUPRT module to do TOTAL COMPATIBILITY work.



CSTXUPRT replaces the Modified TOTINT module's pre-DML call. You should move user logic in the Modified TOTINT for SL2102 to the TPRP exit CSTXUPRT module.

- ◆ TPRPSYVF determines the current resource synchronization method.

Register 13 points to a save work area with the following layout:

```
TPRPSAVE DS    18F    SAVE AREA
TPRPTOTS DS    0F    START OF TOTAL COMPATIBILITY AREAS
```



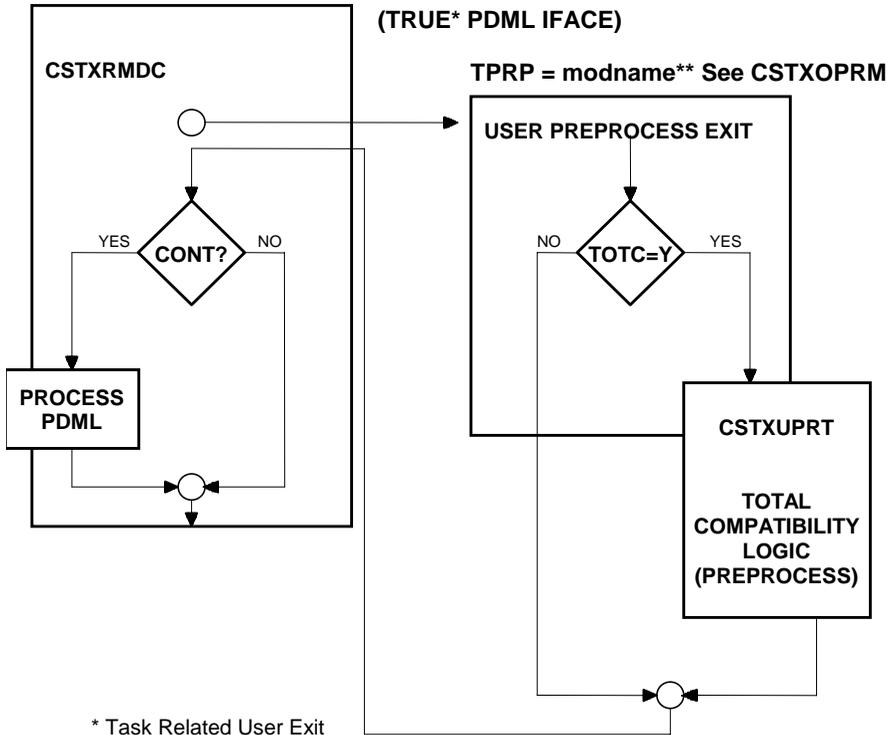
Caution: If CSTXUPRT is called, the subsequent fields are shared across the TPRP exit and CSTXUPRT.

```
TPRPCSVE DS    18F    EXTRA SAVE AREA OR CSTXUSYN SAVE
TPRPWORK DS    18F    USER WORK AREA
TPRPTCOM DS    A      ADDRESS OF 18F USER TASK COMMON AREA
TPRPGADR DS    A      ADDRESS OF 18F USER GLOBAL COMMON AREA
```

- ◆ This is a task-unique area.
- ◆ TPRPTCOM is used for communication between all user exits for a specific task.
- ◆ TPRPGADR is a global area in the global TRUE work area that is identical for all exits using that TRUE work area. It is used for communication between user exits across tasks for the life of the connect.

When you exit, the return code (RC) values in Register 15 are ignored. However, TPRPDMLF is inspected to determine whether the DML should continue.

The following figure illustrates the TPRP user exit.



The TPOP (post-process) user exit

You define the TPOP exit with the TPOP parameter in the CSTXOPRM macro.

- ◆ If you code a user name for TPOP in your CSTXOPRM macro, that user name will be the name of the post-process exit module loaded during OPER CONNECT.

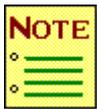


If you code the user name TPOP=CSTXUSER, you can update the TPOP sample logic in the CSTXUSRX sample source to suit your needs. Use the CSTXUSER linkdeck to install the changed CSTXUSER.

- ◆ If you do not code a user name in CSTXOPRM, the macro automatically codes the default name of NO, and the exit address will be set to zero during OPER CONNECT.

Register 1 points first to the data area defined by the C\$XTTPOS macro. (Refer to that macro for a detailed description of all parameters and functions.) The parameters and options directly related to the required TPOP functions are as follows:

- ◆ TPOPDML points to the current PDML parameter address list.
- ◆ TPOPRDMF DML FLAG can be set (YES or NO) to cause the PDML request to be repeated.
- ◆ TPOPTOTF TOTAL COMPATIBILITY can be set to REQUIRED or NOT REQUIRED. If you set it to REQUIRED, CSTXUPOT will be called to do TOTAL COMPATIBILITY work.



CSTXUPOT replaces the Modified TOTINT module's post-DML call. You should move any user logic from the Modified TOTINT for SL2102 to the TPOP exit CSTXUPOT module.

Register 13 points to a save work area with the following layout:

```
TPOPSAVE DS    18F    SAVE AREA
TPOPTOTS DS    0F     START OF TOTAL COMPATIBILITY AREAS
```

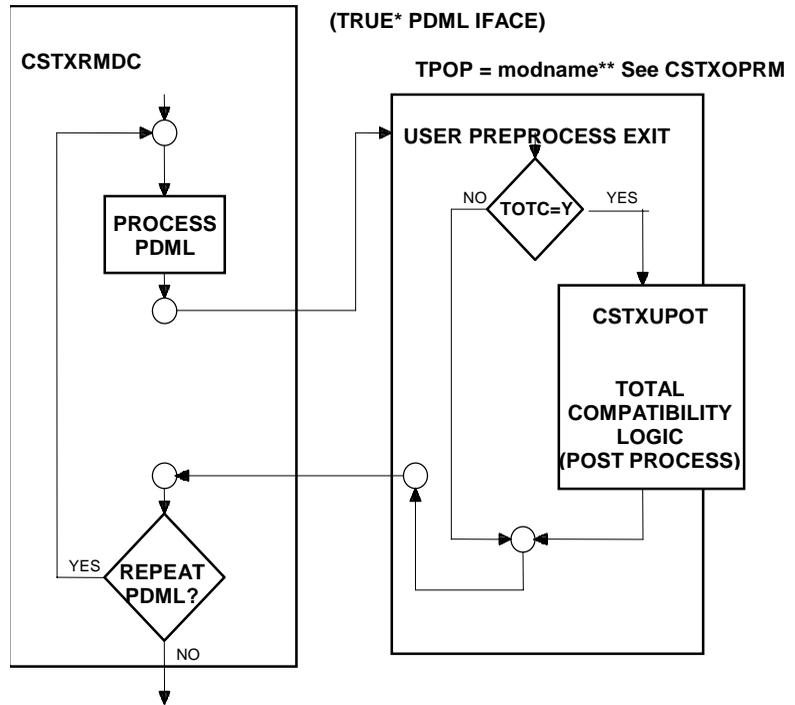


Caution: If CSTXUPOT is called from this exit, the fields beyond this point will be shared across the TPRP exit and CSTXUPRT.

```
TPOPCSVE DS    18     EXTRA SAVE AREA FOR USER CALLS
TPOPWORK DS    18F    USER WORK AREA
TPOPTCOM DS    A      ADDRESS OF 18F USER TASK COMMON AREA
TPOPGADR DS    A      ADDRESS OF 18F USER GLOBAL COMMON AREA
```

- ◆ This is a task-unique area.
- ◆ TPOPTCOM is used for communication between all user exits for a specific task.
- ◆ TPOPGADR is a global area in the global TRUE work area that is identical for all exits using that TRUE work area. It is used for communication between user exits across tasks for the life of the connect. When you exit, the return code (RC) values in Register 15 are ignored. However, TPOPRDMF is inspected to determine if DML should be repeated. If it is repeated, the TPOP exit is reentered after DML processing.

The following figure illustrates the TPOP user exit.



* Task Related User Exit
 ** See CSTXUSER:USERTPOP

The TPRE and TPOS user exits

TPRE and TPOS replace the old CSTXDBXT pre- and post-DML exits.

The TPRE (pre-DML) exit

You define the TPRE exit with the TPRE parameter in the CSTXOPRM macro.

- ◆ If you code a user name for TPRE in your CSTXOPRM macro, that user name will be the name of the pre-DML exit module loaded during OPER CONNECT.
- ◆ If you do not code a user name in CSTXOPRM, the macro automatically codes the default name of NO, and the exit address will be set to zero during OPER CONNECT.

Register 1 points first to the data area defined by the C\$T\$TPRE macro. (Refer to that macro for a detailed description of all parameters and functions.) The parameters and options directly related to the required TPRE functions are as follows:

- ◆ TPREDML points to the current PDML parameter address list.
- ◆ TPRECFRF identifies who called you (a user DFHRMCAL request, an internal PDML from a task exit, etc.).
- ◆ TPRESYVF identifies the current resource synchronization mode.

Register 13 points to a save work area with the following layout:

TPRESAVE DS	18F	SAVE AREA
TPRECSVE DS	18F	EXTRA SAVE AREA FOR USER CALLS
TPREWORK DS	18F	USER WORK AREA
TPRETCOM DS	A	ADDRESS OF 18F USER TASK COMMON AREA
TPREGADR DS	A	ADDRESS OF 18F USER GLOBAL COMMON AREA

- ◆ This is a task-unique area.
- ◆ TPRETCOM is used for communication between all user exits for a specific task.
- ◆ TPREGADR is a global area in the global TRUE work area which is identical for all exits using that TRUE work area.

When you exit, the following conditions apply, depending on the return code (RC) value in Register 15:

- ◆ If RC=4, the DML is passed to the PDM Interface Logic and TPOS is called.
- ◆ If RC is any other value less than 8, the DML is passed to the PDM Interface Logic (TPOS is not called).
- ◆ If RC is greater than 8, control is returned to the user and the DML is not passed to the PDM Interface logic.

The TPOS (post-DML) user exit

You define this exit with the TPOS parameter in the CSTXOPRM macro.

- ◆ If you code a user name for TPOS in your CSTXOPRM macro, that user name will be the name of the post-DML exit module loaded during OPER CONNECT.



If you code the user name TPOS=CSTXUSER, you can update the sample use logic in the CSTXUSRX sample source to suit your needs. Use the CSTXUSER linkdeck to install the changed CSTXUSER.

- ◆ If you do not code a user name in CSTXOPRM, the macro automatically codes the default name of NO, and the exit address will be set to zero during OPER CONNECT.

Register 1 points first to the data area defined by the C\$TXTPOS macro. (Refer to that macro for a detailed description of all parameters and functions.) The parameters and options directly related to the required TPOS functions are as follows:

- ◆ TPOSDML points to the current PDML parameter address list.
- ◆ TPOSCFRF identifies who called you (a user DFHRMCAL request, an internal PDML from a task exit, etc.).
- ◆ TPOSSYVF identifies the current resource synchronization mode.
- ◆ You can set TPOSSYIF to turn on the Task Related User Exit (TRUE) synchronization interest flag.
- ◆ If you set TPOSSYVF equal to TPOSSYVU (user-controlled synchronization), the following conditions apply:
 - TPOSUEP points to DHFUEPAR.
 - You can set TPOSUSYF to invoke CICS SYNC POINT.
 - You can set TPOSURYF to invoke CICS SYNC POINT ROLLBACK.
 - You can set TPOSUSIF to turn on the TRUE synchronization interest flag so that TRUE is invoked at CICS SYNC POINT and CICS SYNC POINT ROLLBACK.

CSTXUSY3 will be called, allowing you to control the resource synchronization. Calling these parameters may affect your save work area. See the caution note in “[The TSYN \(Task Synchronization\) user exit](#)” on page 122.

Register 13 points to a save work area with the following layout:

TPOSSAVE DS 18F SAVE AREA



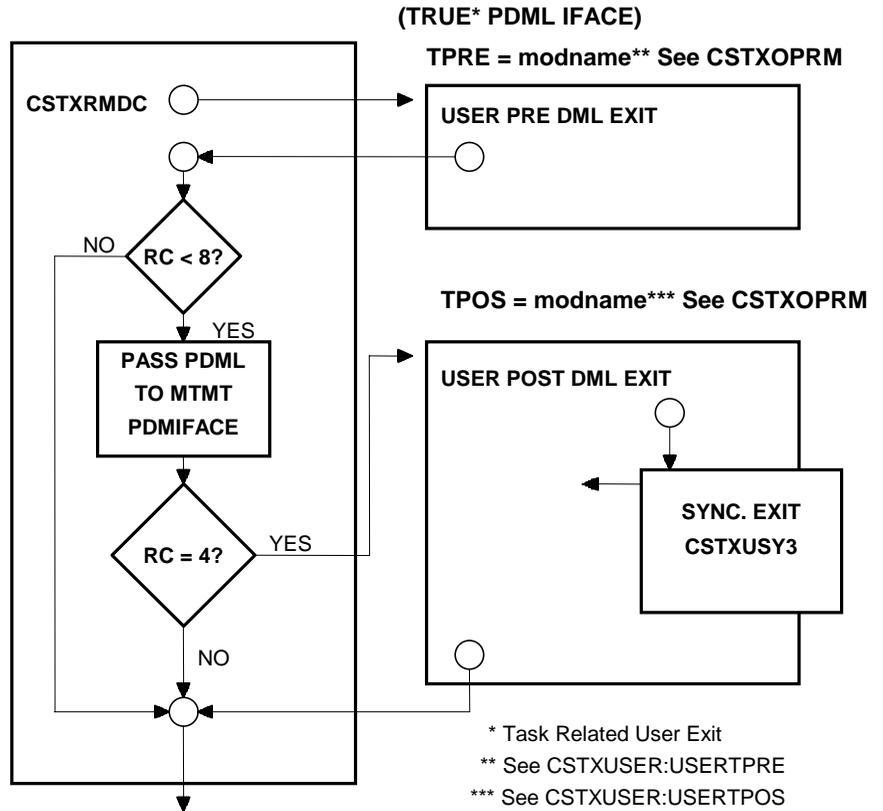
Caution: If CSTXUSY3 is called from this exit, the fields beyond this point will be shared with the TPOS logic. You must design these parameters carefully to ensure that data stored for TPOS is safe.

TPOSUSYS DS	0F	START OF TSYNUSYN (see C\$TXTSYN)
TPOSCSVE DS	18F	EXTRA SAVE AREA, OR TSYNCSVE
TPOSWORK DS	18F	USER WORK AREA
TPOSTCOM DS	A	ADDRESS OF 18F USER TASK COMMON AREA
TPOSGADR DS	A	ADDRESS OF 18F USER GLOBAL COMMON AREA
TPOSUPAD DS	F	PARAMETER ADDRESS LIST AREA

- ◆ This is a task-unique area.
- ◆ TPOSTCOM is used for communication between all user exits for a specific task.
- ◆ TPOSGADR is a global area in the global TRUE work area that is identical for all exits using that TRUE work area. It is used for communication between user exits across tasks for the life of the connect.

When you exit, the return code (RC) values in Register 15 are ignored, as there is no conditional path taken on the post-DML return.

The following figure illustrates the TPRE and TPOS user exits.



The TSYN (Task Synchronization) user exit

You define the TSYN exit with the TSYN parameter in the CSTXOPRM macro.

- ◆ If you code a user name for TSYN in your CSTXOPRM macro, that user name will be the name of the Task Synchronization User Exit module loaded during OPER CONNECT.



If you code the user name TSYN=CSTXUSER, you can update the TSYN sample user logic in the CSTXUSRX sample source to suit your needs. Use the CSTXUSER linkdeck to install the changed CSTXUSER.

- ◆ If you do not code a user name in CSTXOPRM, the macro automatically codes the default name of NO, and the exit address will be set to zero during OPER CONNECT.

Register 1 points first to the data area defined by the C\$TXTSYN macro. (Refer to that macro for a detailed description of all parameters and functions.) The parameters and options directly related to the required TSYN functions are as follows:

- ◆ TSYNSTPF identifies the synchronization request type.
- ◆ TSYNSYVF is the current resource synchronization mode.



If TSYNSYVF = TSYNSYVU, TSYNUEP contains the DFHUEPAR address and TSYNUPAD is used as a parameter address list to call CSTXSYN3.

- ◆ TSYNCOMF is the commit-requested flag. You can reset this flag if you wish.
- ◆ TSYNRSEF is the reset-requested flag. You can reset this flag if you wish.
- ◆ You can also call module CSTXUSY3 during TSYN processing. See the documentation in the MACLIB for more information.

Register 13 points to a save work area with the following layout:

TSYNSAVE DS	18F	SAVE AREA
TSYNUSYN DS	0F	START OF CSTXUSYN R13 AREA



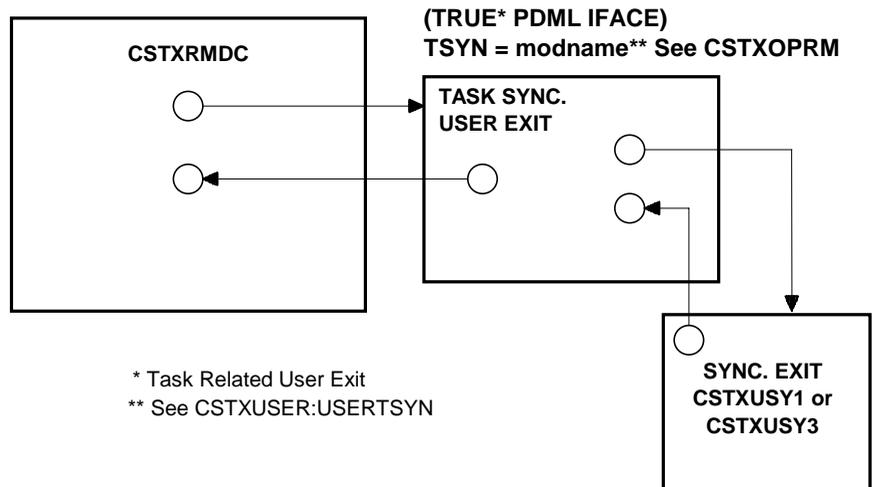
Caution: If CSTXUSY3 is called, the subsequent fields are shared with the TSYN logic.

TSYNSVE DS	18F	EXTRA SAVE AREA OR CSTXUSYN SAVE
TSYNWORK DS	18F	USER WORK AREA
TSYNTCOM DS	A	ADDRESS OF 18F USER TASK COMMON AREA
TSYNGADR DS	A	ADDRESS OF 18F USER GLOBAL COMMON AREA

- ◆ This is a task-unique area.
- ◆ TSYNTCOM is used for communication between all user exits for a specific task.
- ◆ TSYNGADR is a global area in the global TRUE work area that is identical for all exits using that TRUE work area. It is used for communication between user exits across tasks for the life of the connect.

When you exit, the return code (RC) values in Register 15 are ignored.

The following figure illustrates the TSYN user exit.



The TDUE (Task Detach) user exit

You define the TDUE exit with the TDUE parameter in the CSTXOPRM macro.

- ◆ If you code a user name for TDUE in your CSTXOPRM macro, that user name will be the name of the Task Detach User Exit module loaded during OPER CONNECT.



This exit is called only if you are signed on to the PDM at task detach.

If you code the user name TDUE=CSTXUSER, you can update the TDUE sample user logic in the CSTXUSRX sample source to suit your needs. Use the CSTXUSER linkdeck to install the changed CSTXUSER.

- ◆ If you do not code a user name in CSTXOPRM, the macro automatically codes the default name of NO, and the exit address will be set to zero during OPER CONNECT.

The standard logic in the TDUE exit process has the following special features which you may wish to use or disable:

- ◆ If you schedule a reset or sign-off and supply a TMSG transaction ID, TDUE will attempt an EXEC CICS START TRANID(TMSG) NOHANDLE. In the default configuration, this will start TRANSID: TMSG with CSTXTMSG as its program. CSTXTMSG attempts to write a message to the terminal associated with the detaching task, informing it that a reset/sign-off is occurring.

If you do not want TMSG to start, code OCPTRM=NO in the CSTXOPRM macro expansion.

Register 1 points first to the data area defined by the C\$TXTDUE macro. (Refer to that macro for a detailed description of all parameters and functions.) The parameters and options directly related to the required TPOP functions are as follows:

- ◆ TDUETID points to the current task ID.
- ◆ TDUEENTRN points to the next transaction ID. (If this field is X'00000000', no subsequent ID was encountered.)
- ◆ TDUETMSG is the transaction ID for the optional utility program that writes a message to the terminal associated with the detaching task in the event a reset/sign-off is scheduled.
- ◆ TDUERSFG is the Reset/Sinof flag. Use this flag to determine the status of a reset or sign-off. To set this flag, refer to the equate values in C\$TXTDUE.
- ◆ TDUESOTF is a copy of the current Signed-On Table Block (SOTB) flag for the task ID. This flag tells you the pending status of the signed-on task.
- ◆ TDUECFRM is the called-from flag. This flag tells you how the exit gained control—where it was called from.
- ◆ TDUESYVF is the current resource synchronization mode.
- ◆ TDUEABCD is spaces if task is not abending or the CICS task abend code if the task is abending.

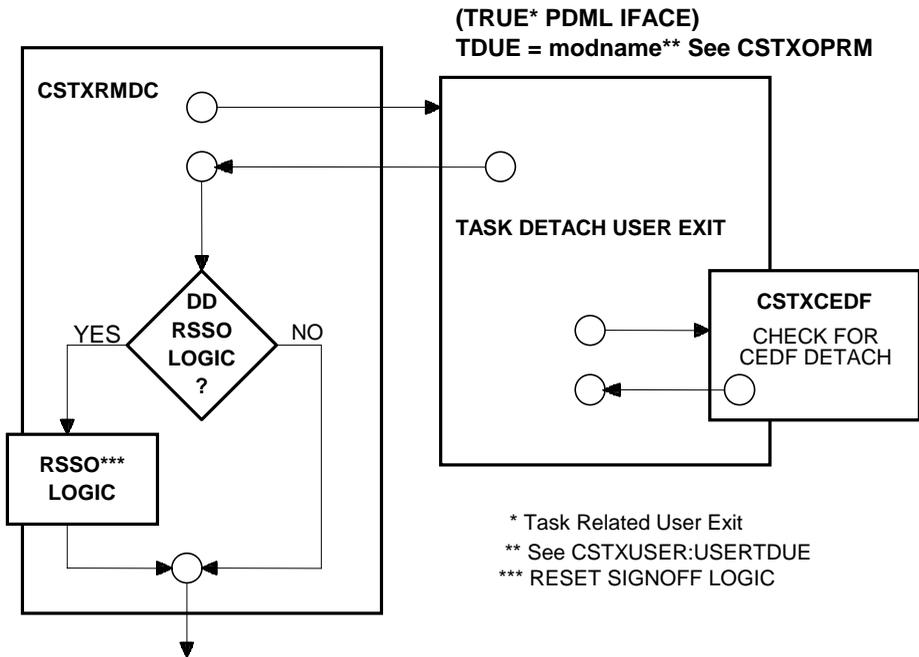
Register 13 points to a save work area with the following layout:

TDUESAVE	DS	18F	SAVE AREA
TDUEUSVE	DS	18F	EXTRA SAVE AREA FOR USER CALLS
TDUEWORK	DS	18F	USER WORK AREA
TDUETCOM	DS	A	ADDRESS OF 18F USER TASK COMMON AREA
TDUEGADR	DS	A	ADDRESS OF 18F USER GLOBAL COMMON AREA

- ◆ This is a task-unique area.
- ◆ TDUETCOM is used for communication between all user exits for a specific task.
- ◆ TDUEGADR is a global area in the global TRUE work area that is identical for all exits using that TRUE work area. It is used for communication between user exits across tasks for the life of the connect.

When you exit, the return code (RC) values in Register 15 are ignored.

The following figure illustrates the TDUE user exit.



The CSTXMRXT exit

You can use the CSTXMRXT exit to control the number of CICS start transactions issued by the LST1/CSTXLST1 transaction/program.

When it receives control, the CSTXLSTX1 program checks whether the CSTXMRXT user exit has been included in the composite link. If it has, the CSTXLST1 program initializes all transaction indicator fields to Y to start the transaction and then passes control to the user exit.

The following four registers are passed to CSTXMRXT:

- ◆ Register 1 points to the parameter list with the following addresses:
 - +0 contains the address of the field that contains the 8-character terminal ID.
 - +4 contains the address of the start of the CSTXMROT Table. Each entry consists of 16 bytes, as follows:
 - 4-byte transaction ID
 - 1-byte transaction indicator (initialized to Y)
 - 11-byte comment field
- ◆ Register 13 points to an 18 full word save area.
- ◆ Register 14 contains the return address.
- ◆ Register 15 contains the entry point address of CSTXMRXT.

If you do not want the CICS START transaction started for a particular transaction, change the indicator to something other than Y.

When the 4-byte transaction ID contains X'FFFFFFFF,' you have reached the end of the CSTXMROT table. At this point, you should return control to the caller, after restoring the registers.

This exit is available only to users who can control which terminals go to which CICS application-owning regions by “generic” means. A single terminal can perform several transactions from different CICS regions simultaneously, and the only CICS region that is notified of any problem is the region that “owns” the terminal. Consequently, the LST1 transaction and the CSTXMROT table are designed to resolve problems caused by terminals going out of service during a pseudoconversational hiatus. Do not perform any external calls from this user exit. The CICS transaction LST1 is in control, and you do not have any dynamic DSA other than the 18 full words that Register 13 points to. If you use static DSA, the results will be unpredictable.

The MACLIB contains a stub CSTXMRXT.

Obsolete user exits

The exits discussed in this section are no longer used. In most cases, exits discussed earlier in this chapter incorporate the functionality of these obsolete exits:

1. XPCFTCH. SUPRA Server 1.3.x uses this exit to place the interface addresses in the user copy of DATBASC. This function is now handled by the DFHRMCAL macro coded directly into DATBASC. You can create your own XPCFTCH exit if you wish. Refer to the *IBM CICS Customization Guide* for more information.
2. XDBINIT. SUPRA Server 1.3.x uses this exit to reset or sign off abending tasks that were signed on to the SUPRA PDM. The RMI now handles this function at task detach.

This exit also had the auxiliary function of executing a SUPRA RESET DML if synchronization was indicated during a CICS ROLLBACK. The RMI TRUE Synchronization exit now handles this function.

If you wish to supply your own XDBINIT exit, refer to the *IBM CICS Customization Guide* for more information.

3. XKCREQ. SUPRA Server 1.3.x uses this exit to reset or sign off detaching tasks which are signed on to the SUPRA PDM and which either have uncommitted updates or are not pseudoconversational. The RMI now performs this function at task detach.
4. Modified TOTINT: TOTAL Compatibility Exit

Modified TOTINT is obsolete and has been replaced by the [TPRP and TPOP exits](#).

5. CSTXDBXT. The CSTXDBXT pre-/post-command exit has been replaced by the [TPRE and TPOS pre- and post-command exits](#).

If you wish to intercept abending tasks during CICS Dynamic Task Backout, you can implement your own XDBINIT exit. Refer to the *IBM CICS Customization Guide* for more information.

6. CSTXKCCX. The Task Detach user exit has been replaced by the [TDUE \(Task Detach User Exit\)](#).

7

Using Cincom-supplied transactions

Overview of supplied transactions

Transactions are now being shipped with the CICS connector for getting information from the connected PDM. Current information available is file and connector related. Six CICS transactions are supplied (ISRA, ISRB, ISRC, ISRD, ISRE, and ISRF). The first three characters of these transactions can be modified by using the INFOTRAN parameter of the CSTXOPRM macro expansion. Each transaction is explained below.

ISRA PDM information menu

Transaction ISRA is a menu transaction that will allow selection of one of the five menu options. A sample screen is shown below.

```
ISRA                CONNECTOR SERVICES MENU
                   SELECT THE DESIRED SERVICE

                   FILE INFORMATION
                   FILE STATISTICS
                   CONNECTOR INFORMATION
                   PDM INFORMATION
                   SECONDARY KEY INFORMATION
```

ISRC PDM File Information

Transaction ISRC can be selected from transaction ISRA or entered directly. ISRC will first display a list of all files that are available in the current PDM. From this list you can select one or more files with I for information or F for file elements. Information about the selected files is then displayed.

ISRD PDM File usage Information

Transaction ISRD can be selected from transaction ISRA or entered directly. ISRD will first display a list of all files that are available in the current PDM. From this list you can select one or more files with S for statistics. Usage information about the selected files is then displayed.

ISRF PDM Connector Information

Transaction ISRF can be selected from transaction ISRA or entered directly. ISRF will first display a list of all interfaces connected to the current PDM. From this list you can select an interface for which you need more information. If you select a CICS type interface the following information is displayed about each task currently connected via this interface.

- ◆ Task name
- ◆ Task status
- ◆ Time and date of last commit
- ◆ Task restart flag
- ◆ CICS signed on table information. As (signed on status/inflight status/updates status/rso status). ! means not, so a response of ON!/ACTIVE!/UPDATES!/RSSO would mean that this task is signed on to the PDM, is not currently active in the PDM, has no uncommitted updates, and is not marked for forced reset/sinof. A response of ON!/ACTIVE/UPDATES!/RSSO would indicate that the task is signed on, is not currently active in the PDM, has uncommitted updates, and is not marked for forced reset/sinof.

From this list of tasks you may request that a task be purged from the PDM. To do this tab to the task you wish to purge and select with any non blank character and press enter. This purge is accomplished by starting transaction LST1 passing the terminal/task name the same as DFHZNEP does with Vtam errors.

ISRB PDM Information

Transaction ISRB can be selected from transaction ISRA or entered directly. ISRB displays the current PDM Information.

ISRE Secondary Key Information

Transaction ISRE can be selected from transaction ISRA or entered directly. ISRE will first display a list of all secondary keys that are available in the current PDM. From this list you can select one or more secondary keys with I for information or F for file elements that make up the secondary key. Information about the selected secondary key is then displayed.

8

Cincom-supplied source programs

The following programs are available in source and executable versions. They are released in source so the user may modify the functioning as needed for their specific environment. Under most conditions they will function correctly as shipped.

CSTXGNIT

This program is an example security time out exit. As coded it will start transaction LST1 to force a reset/sinof of the timed out task. For this program to be active you must supply a CICS program definition, a CICS transaction for GNIT, and a SIT parameter or SIT override of GNTRAN=GNIT . This program is also on the SUPRA linklib in a executable version.

CSTXLST1

This program has been discussed already under “[Tailoring NEP for SUPRA](#)” on page 76.

CSTXUSRX

This program has been discussed already in “[Using Cincom CICS Connector exits](#)” starting on page 99.

DFHZNEP

This program has been discussed already under “[Tailoring NEP for SUPRA](#)” on page 76.

A

Using Cincom CICS Connector CSTXOPRM macro

Assembling the CSTXOPRM macro

The Operator Control Parameter Table macro CSTXOPRM builds the CICS operator control parameter table CSTXOTBL. To assemble this macro, see CSTXOTBL, a member on the distributed source statement library for the CSTXOPRM assembly default used in creating the distributed CSTXOTBL module. Also see the JCL sample TXJAOPRM.

This macro has three formats:

- ◆ **Format 1** is required to define the default CONNECT values.
- ◆ **Format 2** is used to group files.
- ◆ **Format 3** is used to define files.

Format 1 (define default CONNECT values)

CSTXOPRM TABLE	X
$\left[, \text{ALOWHLD} = \left\{ \begin{array}{c} \text{Y} \\ \text{N} \end{array} \right\} \right]$	X
$\left[, \text{CFUL} = \left\{ \begin{array}{c} \underline{30} \\ \text{nnnnn} \end{array} \right\} \right]$	X
$\left[, \text{ICOR} = \left\{ \begin{array}{c} \underline{5} \\ \text{nnnnn} \end{array} \right\} \right]$	X
$\left[, \text{IFRTRY} = \left\{ \begin{array}{c} \underline{5} \\ \text{nnn} \end{array} \right\} \right]$	X
$\left[, \text{INFOPRE} = \left\{ \begin{array}{c} \underline{\text{ISR}} \\ \text{xxx} \end{array} \right\} \right]$	X
$\left[, \text{MAXPACK} = \left\{ \begin{array}{c} \underline{0} \\ \text{nnnnn} \end{array} \right\} \right]$	X
$\left[, \text{MBLOC} = \left\{ \begin{array}{c} \underline{\text{ABOVE}} \\ \underline{\text{BELOW}} \end{array} \right\} \right]$	X
$\left[, \text{MBSIZE} = \left\{ \begin{array}{c} \underline{65504} \\ \text{nnnnn} \end{array} \right\} \right]$	X
$\left[, \text{MEME} = \left\{ \begin{array}{c} \underline{5} \\ \text{nn} \end{array} \right\} \right]$	X

$\left[,\text{MESSQID} = \left\{ \begin{array}{l} \text{MUSR} \\ \text{xxxx} \end{array} \right\} \right]$	X
$\left[,\text{OCPTRID} = \left\{ \begin{array}{l} \text{OPER} \\ \text{xxxx} \end{array} \right\} \right]$	X
$\left[,\text{OCPTRTM} = \left\{ \begin{array}{l} \text{NO} \\ \text{xxxx} \end{array} \right\} \right]$	X
$\left[,\text{PUSHPOP} = \left\{ \begin{array}{l} \text{N} \\ \text{Y} \end{array} \right\} \right]$	X
$\left[\text{SYNC} = \left\{ \begin{array}{l} \text{N} \\ \text{U} \\ \text{C} \end{array} \right\} \right]$	X
$\left[,\text{TASKS} = \left\{ \begin{array}{l} \text{3} \\ \text{nnnnn} \end{array} \right\} \right]$	X
$\left[,\text{TDUE} = \left\{ \begin{array}{l} \text{NO} \\ \text{modname} \end{array} \right\} \right]$	X
$\left[,\text{TFUL} = \left\{ \begin{array}{l} \text{30} \\ \text{nnnnn} \end{array} \right\} \right]$	X
$\left[,\text{THREADS} = \left\{ \begin{array}{l} \text{2} \\ \text{nnnnn} \end{array} \right\} \right]$	X
$\left[,\text{TOTC} = \left\{ \begin{array}{l} \text{N} \\ \text{Y} \end{array} \right\} \right]$	X

$\left[,TPOP = \left\{ \begin{array}{l} \underline{NO} \\ \textit{modname} \end{array} \right\} \right]$	X
$\left[,TPOS = \left\{ \begin{array}{l} \underline{NO} \\ \textit{modname} \end{array} \right\} \right]$	X
$\left[,TPRE = \left\{ \begin{array}{l} \underline{NO} \\ \textit{modname} \end{array} \right\} \right]$	X
$\left[,TPRP = \left\{ \begin{array}{l} \underline{NO} \\ \textit{modname} \end{array} \right\} \right]$	X
$\left[,TRACE1 = \left\{ \begin{array}{l} \underline{5} \\ \textit{nnn} \end{array} \right\} \right]$	X
$\left[,TRACE2 = \left\{ \begin{array}{l} \underline{6} \\ \textit{nnn} \end{array} \right\} \right]$	X
$\left[,TRACE3 = \left\{ \begin{array}{l} \underline{7} \\ \textit{nnn} \end{array} \right\} \right]$	X
$\left[,TSYN = \left\{ \begin{array}{l} \underline{NO} \\ \textit{modname} \end{array} \right\} \right]$	X
$\left[,TUID = \left\{ \begin{array}{l} \underline{NO} \\ \textit{modname} \end{array} \right\} \right]$	X
$\left[,USERTID = \left\{ \begin{array}{l} \underline{NO} \\ \textit{xxxx} \end{array} \right\} \right]$	X

TABLE

Restriction	OS/390 and VSE only.
Description	<i>Required.</i> Identifies this as defining the default CONNECT values.
Consideration	You can code only one TABLE entry for CSTXOTBL in a source module. The TABLE entry must be the first entry coded and must precede the GROUP and FILE entries.

ALOWHLD= $\left. \begin{array}{c} \text{Y} \\ \text{N} \end{array} \right\}$

Description	<i>Optional.</i> Specifies whether task will be allowed to go Pseudo with held records.
Default	N
Options	Y Yes N No

Considerations

- ◆ Previous releases would allow tasks to go Pseudo with held records. Tasks were not allowed to go Pseudo with uncommitted updates.
 - ◆ If you code N tasks with held records will be treated the same as tasks with uncommitted updates.
-

CFUL= $\left. \begin{array}{c} \underline{30} \\ nnnnn \end{array} \right\}$

Description	<i>Optional.</i> Defines the number of retries (with a one-second delay) after a CFUL status.
Default	30
Format	1-32767
Consideration	Tasks and threads can be set to average values and retry values can be set high to handle spikes in demand.

ICOR= $\left\{ \begin{array}{c} \underline{5} \\ nnnn \end{array} \right\}$

Description *Optional.* Defines the number of retries (with a one-second delay) after an ICOR status.

Default 5

Format 1–32767

Consideration To correct ICOR problems, a reanalysis of resource requirements is usually required.

IFRTRY= $\left\{ \begin{array}{c} \underline{5} \\ nnn \end{array} \right\}$

Description *Optional.* Specifies the number of one-second delays to issue while waiting to free task context for an inflight PDM request.

Default 5

Options 1–999

Considerations

- ◆ IFRTRY should be set to the maximum return time for a single DML.
 - ◆ IFRTRY should always be set greater than the PDM environment descriptor parameter TP-MONITOR Delay Time.
-

INFOPRE= $\left\{ \begin{array}{c} \underline{ISR} \\ xxx \end{array} \right\}$

Description *Optional.* Specifies the first 3 characters of the transactions used to define the connector information system to CICS.

MAXPACK= $\left\{ \begin{array}{l} \underline{0} \\ nnnnn \end{array} \right\}$

Restriction VSE only

Description *Optional.* Documents the maximum packet size for a parameter.

Default 0

Options 1–37766

MBLOC= $\left\{ \begin{array}{l} \underline{\text{ABOVE}} \\ \underline{\text{BELOW}} \end{array} \right\}$

Description *Required.* Specifies whether shared connector memory comes from above or below the line CICS DSA.

Default ABOVE

MBSIZE= $\left\{ \begin{array}{l} \underline{65504} \\ nnnnn \end{array} \right\}$

Description *Required.* Specifies pool size for CICS connector shared memory.

Default 65504

Options 64528-65504

Considerations

- ◆ Use the default for OS/390.
- ◆ For VSE CICS 2.3 use 65400.

$$\text{MEME} = \left\{ \begin{array}{c} \underline{5} \\ nn \end{array} \right\}$$

- Description** Optional. Specifies the maximum number of additional memory blocks that can be acquired if memory allocation is exceeded during the connect.
- Default** 5
- Options** 1–99
- Consideration** This should be large enough to allow for some expansion but small enough to prevent excessive memory allocation without warning. In VSE environments, consider **MAXPACK** times the number of concurrent tasks doing DML divided by the block size (65504).
-

$$\text{MESSQID} = \left\{ \begin{array}{c} \underline{\text{MUSR}} \\ \text{xxxx} \end{array} \right\}$$

- Description** *Optional.* Defines the queue ID in your DCT table for the audit log.
- Default** MUSR
- Format** 4 alphanumeric characters
-

$$\text{OCPTRID} = \left\{ \begin{array}{c} \underline{\text{OPER}} \\ \text{xxxx} \end{array} \right\}$$

- Description** *Optional.* Specifies the transaction ID for the operator control program.
- Default** OPER
- Format** 4 alphanumeric characters
- Consideration** Must be the same as the TRANSID parameter on the PCT entry for program CSTXOPER.

OCPTRTM= $\left\{ \begin{array}{l} \text{NO} \\ \text{xxxx} \end{array} \right\}$

Description *Optional.* Specify NO or the transaction ID of a custom user program that will be linked at the end of RESET/SINOF processing. Supplied program CSTXTMSG (TMSG) can be used to notify the operator that a task is being reset or signed off (RESET/SINOF).

Default NO

Format NO or 4 alphanumeric characters

Consideration The transaction definition must reference CSTXTMSG or a custom user program.

PUSHPOP= $\left\{ \begin{array}{l} \text{N} \\ \text{Y} \end{array} \right\}$

Description *Optional.* Handles the PUSH/POP option for CICS commands. It allows internally generated EXEC CICS requests to be bracketed with EXEC CICS PUSH HANDLE-CICS POP HANDLE condition requests.

Default N

Consideration You may need this if user exits set handle conditions.

SYNC= $\left. \begin{array}{c} \text{N} \\ \text{U} \\ \text{C} \end{array} \right\}$

- Description** *Optional.* Handles the task synchronization option.
- Default** N
- Options** N No
- U User defined. Activates CSTXUSY3 to allow the user to control sync actions.
- C Prior compatibility. SUPRA commit generates a CICS sync point, and a CICS sync point rollback generates a SUPRA reset.

Considerations

- ◆ Should be set to the minimum required by your resources.
- ◆ Set to C only to keep CICS resources in recoverable sync with SUPRA PDM resources.
- ◆ TPOS=CSTXUSER must be coded if option C or U is coded.

TASKS= $\left. \begin{array}{c} 3 \\ nnnnn \end{array} \right\}$

- Description** *Optional.* Specifies the default number of CICS tasks that may be signed on to the PDM at any time.
- Default** 3
- Format** 1–32,767

Considerations

- ◆ This is the default value for the TASKS parameter on the CONNECT operator control command.
- ◆ This affects the number of bytes of virtual storage used. The more tasks specified, the more memory required.

TDUE= $\left\{ \begin{array}{l} \underline{\text{NO}} \\ \text{modname} \end{array} \right\}$

Description *Optional.* Specifies the user exit name. CSTXUSER supplies the default logic for this exit point.

Default NO

Considerations

- ◆ The modname cannot be NO.
 - ◆ The modname is CSTXUSER unless you want to separate the exits into different modules, or you have existing modules you want to use.
-

TFUL= $\left\{ \begin{array}{l} \underline{30} \\ \text{nnnnn} \end{array} \right\}$

Description *Optional.* Defines the number of retries (with a one-second delay) after a TFUL status.

Default 30

Format 1–32767

Consideration Tasks and threads can be set to average values and retry values can be set high to handle spikes in demand.

THREADS= $\left\{ \begin{array}{l} \underline{2} \\ \text{nnnnn} \end{array} \right\}$

Description *Optional.* Specifies the default number of threads between the CICS connector and the PDM. This is the default for the THREADS parameter on the CONNECT operator control command.

Default 2

Options 1–32767

TOTC= $\left\{ \begin{array}{c} \text{N} \\ \text{Y} \end{array} \right\}$

Description *Optional.* Provides TOTAL compatibility for SUPRA DML.

Default N

Consideration If set to Y then TPRP=CSTXUSER and TPOP=CSTXUSER must also be coded.

TPOP= $\left\{ \begin{array}{c} \text{NO} \\ \text{modname} \end{array} \right\}$

Description *Optional.* Specifies the user exit name. CSTXUSER supplies the default logic for this exit point.

Default NO

Considerations

- ◆ The modname cannot be NO.
 - ◆ The modname is CSTXUSER unless you want to separate the exits into different modules, or you have existing modules you want to use.
-

TPOS= $\left\{ \begin{array}{c} \text{NO} \\ \text{modname} \end{array} \right\}$

Description *Optional.* Specifies the user exit name. CSTXUSER supplies the default logic for this exit point.

Default NO

Considerations

- ◆ The modname cannot be NO.
- ◆ The modname is CSTXUSER unless you want to separate the exits into different modules, or you have existing modules you want to use.

TPRE= $\left\{ \begin{array}{l} \text{NO} \\ \text{modname} \end{array} \right\}$

Description *Optional.* Specifies the user exit name. CSTXUSER supplies the default logic for this exit point.

Default NO

Considerations

- ◆ The modname cannot be NO.
- ◆ The modname is CSTXUSER unless you want to separate the exits into different modules, or you have existing modules you want to use.

TPRP= $\left\{ \begin{array}{l} \text{NO} \\ \text{modname} \end{array} \right\}$

Description *Optional.* Specifies the user exit name. CSTXUSER supplies the default logic for this exit point.

Default NO

Considerations

- ◆ The modname cannot be NO.
- ◆ The modname is CSTXUSER unless you want to separate the exits into different modules, or you have existing modules you want to use.

$$\text{TRACE1} = \left\{ \begin{array}{l} 5 \\ nnn \end{array} \right\}$$

Description *Optional.* Specifies the trace ID for PDML commands on entry to the interface.

Default 5

Format 1–199

Considerations

- ◆ You turn tracing off by coding TRACE=, in the CSTXOPRM macro expansion.
- ◆ If you supply a value for TRACE1, tracing is turned on. TRACE2 and TRACE3 entries will be generated automatically using defaults unless you supply values for them.
- ◆ Do not set TRACE1 to zero.
- ◆ SUPRA Server is delivered with tracing turned on in the default module CSTXOTBL.
- ◆ If CICS global tracing is turned off then TRACE1=nnn is ignored and connector tracing is not done.

$$\text{TRACE2} = \left\{ \begin{array}{l} 6 \\ nnn \end{array} \right\}$$

Description *Optional.* Specifies the trace ID for a PDML command status upon exit from the interface.

Default 6

Format 1–199

TRACE3= $\left\{ \begin{array}{l} \underline{7} \\ \mathit{nnn} \end{array} \right\}$

Description *Optional.* Specifies the trace ID for a new pseudo conversational task associated with a PDM task signed on by a prior task (e.g., CICS Task 1 signs on, processes, and passes a next transid to CICS Task 2. When CICS Task 2 starts, the TRACE3 ID displays).

Default 7

Format 1–199

TSYN= $\left\{ \begin{array}{l} \underline{\text{NO}} \\ \mathit{modname} \end{array} \right\}$

Description *Optional.* Specifies the user exit name. CSTXUSER supplies the default logic for this exit point.

Default NO

Considerations

- ◆ The modname cannot be NO.
 - ◆ The modname is CSTXUSER unless you want to separate the exits into different modules, or you have existing modules you want to use.
-

TUID= $\left\{ \begin{array}{l} \underline{\text{NO}} \\ \mathit{modname} \end{array} \right\}$

Description *Optional.* Specifies the user exit name. CSTXUSER supplies the default logic for this exit point.

Default NO

Considerations

- ◆ The modname cannot be NO.
- ◆ The modname is CSTXUSER unless you want to separate the exits into different modules, or you have existing modules you want to use.

USERTID= { NO }
 { **xxxx** }

Description Defines the transaction ID for the user initialization program.

Default NO

Format 4 alphanumeric characters

Considerations

- ◆ A transaction ID cannot be NO.
- ◆ You can use this to cause an EXEC CICS start transaction ID (xxxx) to start a user program that requires a connected PDM.
- ◆ User initialization is started as soon as OPER CONNECT gets a good status from PDML CNNCT.
- ◆ You can use this user initialization program to do any initialization that must wait until the connect is completed.

Format 2 (group files)

CSTXOPRM **GROUP,GROUPNM=gggggggg**

GROUP

Restriction	OS/390 and VSE only
Description	<i>Required.</i> Identifies this as defining the name of the group of files.
Format	Enter as shown.

GROUPNM=gggggggg

Restriction	OS/390 and VSE only
Description	<i>Required.</i> Specifies the name of the group. This value is used in the GROUP parameter of the Operator Control OPEN and CLOSE commands to specify a group of files to be opened or closed.
Format	1–8 alphanumeric characters

Considerations

- ◆ Each group in the CSTXOTBL must have a unique name.
- ◆ A GROUP statement must precede the FILE statements in the group.

Format 3 (define files)

CSTXOPRM	FILE	X
	$\left[,\text{CLSMODE} = \left\{ \begin{array}{c} \text{COMP} \\ \text{PART} \end{array} \right\} \right]$	X
	,FILENM=ffff	X
	$\left[,\text{OPNMODE} = \left\{ \begin{array}{c} \text{SUPD} \\ \text{IUPD} \\ \text{READ} \end{array} \right\} \right]$	
	[LABELN=ffff]	

FILE

Restriction	OS/390 and VSE only
Description	<i>Required.</i> Identifies this as defining the file to be opened or closed.

CLSMODE= $\left\{ \begin{array}{c} \text{COMP} \\ \text{PART} \end{array} \right\}$

Restriction	OS/390 and VSE only	
Description	<i>Optional.</i> Specifies the mode to be used when the file is closed. The file is closed as part of the file group.	
Default	COMP	
Options	COMP	Use the operating system close. The file is unlocked if necessary. The file mode is reset to null. NO task may access the file except to open it.
	PART	Do not use the operating system close. The file is unlocked by CLOSX (nontask-logging) or COMMIT/SIGN-OFF (task-logging), if necessary. The file mode is reset to IUPD. Any task may read the file, but no task may update the file.

FILENM=ffff

Restriction	OS/390 and VSE only
Description	<i>Required.</i> Specifies the name of a PDM file. The same file may appear in more than one group.
Format	4 alphanumeric characters
Consideration	This value must be the name of a file defined on the Directory.

OPNMODE= $\left\{ \begin{array}{l} \text{SUPD} \\ \text{IUPD} \\ \text{READ} \end{array} \right\}$

Restriction	OS/390 and VSE only						
Description	<i>Optional.</i> Specifies the mode to be used when the file is opened. The file is opened as part of the file group.						
Default	SUPD						
Options	<table> <tr> <td>SUPD</td> <td>Marks the file for shared updating. The file is locked and may be updated by all tasks using the same copy of the PDM.</td> </tr> <tr> <td>IUPD</td> <td>Marks the file for read only. The file need not be locked. The file may be read by all tasks but not updated by any task.</td> </tr> <tr> <td>READ</td> <td>Marks the file for read only. The file cannot be locked. The file may be read by any task but not updated.</td> </tr> </table>	SUPD	Marks the file for shared updating. The file is locked and may be updated by all tasks using the same copy of the PDM.	IUPD	Marks the file for read only. The file need not be locked. The file may be read by all tasks but not updated by any task.	READ	Marks the file for read only. The file cannot be locked. The file may be read by any task but not updated.
SUPD	Marks the file for shared updating. The file is locked and may be updated by all tasks using the same copy of the PDM.						
IUPD	Marks the file for read only. The file need not be locked. The file may be read by all tasks but not updated by any task.						
READ	Marks the file for read only. The file cannot be locked. The file may be read by any task but not updated.						

LABELN=ffff

Description	<i>Optional.</i> Specifies the name of a label used in the CSTXOFLE CSECT expansion of the CSTXOPRM file format.
Default	Value of FILENM parameter
Format	4 alphanumeric characters
Consideration	If the FILENM parameter does not define a valid Assembler label, use this parameter to provide a valid pseudonym.

B

Using Cincom CICS Connector C\$TXUNCP macro

This module is used to generate a list of CICS transactions that will be allowed to return to CICS with a next transaction id and uncommitted PDM updates. Under normal processing a transaction will be forced off the PDM if this is attempted.

Assembling the C\$TXUNCP macro

To assemble this macro, see C\$TXUNCP, a member on the distributed source statement library for the C\$TXUNCP assembly default used in creating the distributed C\$TXUNCP module.

The distributed module allows CEDF (CICS trace), and CSPK (CICS PA? Print facility) transactions to go pseudo/terminate with uncommitted updates.

If you modify this module you will need to relink CSTXRMDC using the CSTXRMDC linkdeck.

Index

O

0998 abend code: 54, 59

A

abend codes 54
 0998 54
 D998 54
ABEND command
 description 84
 Operator Control Task 32
 resolving errors 54
 terminating connection 59
abend on detach 29
abends
 concurrent 60
 recovery 60
ACTV status 48
address spaces
 attached central operating
 mode 44
 attached operating mode 43
 central operating mode 41
ALOWHLD parameter,
 CSTXOPRM 139
application programs
 communication with 18
 services for 17
ATTACH parameter, CSIPARM
 file 46
attached central operating mode
 44
attached operating mode 43
audit trail 56
 ID 56
 routing to hardcopy printer 65
audit trail. *See also* trace table
automatic connect 23, 43, 46
automatic disconnect 23, 47, 69

B

Basic Mapping Support (BMS) 61
BMS page command, warning 22
BMS. *See* Basic Mapping
 Support
BSAM sequential terminal 24

C

C\$TXCMUE macro 102
C\$TXMROT macro 102
C\$TXTDUE macro 102
C\$TXTPOP macro 102
C\$TXTPOS macro 102
C\$TXTPRE macro 103
C\$TXTPRP macro 103
C\$XTSYN macro 103
C\$TXUNCP macro, assembling
 155
C\$TXUTID macro 103
C\$VOOPTM macro 55, 74
CEMT P SHUT 23, 24
central operating mode 41, 42
CFUL parameter, CSTXOPRM
 139
CFUL status 31, 49
 retries 89, 139
CFUL, status 73
CICS
 abends, recovery from 60
 connector activity audit trail 65
 defined 17
 defining transactions 67
 Destination Control Table
 (DCT) 63, 64, 65
 DFHSIT parameters 55
 Directory Maintenance tasks 40
 environment setup problems 55
 exit programs 33, 99
 MANTIS applications 39
 Node Error Program (NEP) 28
 operator commands 46
 Program List Table (PLT) 68
 RDM communication 35
 SPECTRA tasks 37
 synchronization 24
 System Initialization Table (SIT)
 70
 task abends 54
 tasks 19
 defined 18
 memory 27

- CICS (*cont.*)
 - termination 23
 - trace table 58
 - Transaction List Table (XLT) 70
 - transactions
 - defined 18
 - types 21
- CICS Connector
 - activity audit trail 33, 48, 56
 - concurrent tasks 19
 - connecting 46
 - defined 17
 - disconnecting 47
 - installing 61
 - macros 102
 - operating 45
 - overview 18
 - resources 62
 - status 49
 - storage allocation 26
 - task flow to PDM 25
- CICSTBLS data set 63
- CICSTBLS library 68
- CLOSE command
 - description 85
 - Directory files 51
 - dynamically closing files 50
 - exceptions 51
 - Operator Control Task 32
- CLSMODE parameter,
 - CSTXOPRM 152
- COMIT command 19
- Common User Exit PARM and
 - work area 102
- communication
 - connecting to the PDM 23
- component tasks, memory 26
- components, relationships
 - between 34
- concurrent tasks 19
- conflicting parameters 55
- CONNECT command
 - concurrent task monitoring 19
 - connecting to PDM 46
 - description 87
 - initiating communication 23
 - Operator Control Task 32
- connecting
 - memory allocation 26
 - connecting the PDM 23
- Connector *See* CICS Connector
- conversational transactions 21
- correspondence, CICS and PDM
 - tasks 20
- CSGAQRUN SPECTRA module 38
- CSIPARM file 46
- CSTX0791 message 92
- CSTXCSMT program 56
- CSTXDBXT user exit 128
- CSTXGNIT 133
- CSTXKCCX user exit 128
- CSTXLST1 133
- CSTXLST1 program 29
- CSTXMRXT
 - exit point 82
- CSTXMRXT user exit 127
- CSTXOPRM macro 72, 113, 116,
 - 122, 124
 - additional memory 27
 - ALOWHLD 139
 - assembling 135
 - CFUL 139
 - CLSMODE 152
 - FILE 152
 - FILENM 153
 - GROUP 151
 - GROUPNM 151
 - ICOR 140
 - IFRTRY 140
 - INFOPRE 140
 - LABELN 153
 - listing current parameters 49
 - MAXPACK 141
 - MBLOC 141
 - MBSIZE 141
 - MEME 142
 - MESSQID 142
 - OCPTRID 142
 - OCPTRTM 143
 - OPNMODE 153
 - PUSHPOP 143
 - SYNC 144
 - TABLE 139
 - TASKS 144
 - TDUE 145
 - TFUL 145
 - THREADS 145
 - TOTC 146
 - TPOP 146
 - TPOS 146
 - TPRE 147
 - TPRP 147
 - TRACE1 148

- CSTXOPRM macro (*cont.*)
 TRACE2 148
 TRACE3 149
 TSYN 149
 TUID 149
 USERTID 150
- CSTXOTBL 72
 changing values 55
- CSTXOTBL. *See* Operator Control Parameter Table
- CSTXPLCI
 activating OPER CONNECT 24
- CSTXPLDT
 activating OPER DISCONNECT 24
- CSTXTMSG sample program 30
- CSTXUDAT user exit 99, 107, 109
- CSTXUSRX 133
- CSTXUSY3 119, 122
- CSTXUTID user exit 104
- CSVICICS interface module 35
- CSVNICIC interface module 35
- Customer Information Control System *See* CICS
- D**
- D998 abend code 54
- Database Administrator (DBA) 40, 85
- database files, dynamic opening/closing 50
- DATBAS
 memory allocated 27
 number of tasks issuing requests 26
 requests from applications 24
 SINON PDML 28
- DCT. *See* Destination Control Table (DCT)
- debugging 53
- debugging tools 56
 ABEND command 59
 ABEND operator control commands 53, 84
 activity audit trail 56
 CICS trace table entries 58
 DUMP command 59
 DUMP operator control command 54, 93
 default values, specifying 46
 Define MRO Remote transactions 102
 delays, number of 140
 Destination Control Table (DCT) 56, 63, 64, 65
 queue ID 142
 detaching 29, 30
 DFHDCT 63
 DFHPLTCI 63
 DFHPLTDT 63
 DFHSIT 55, 61, 63
 DFHXL 63
 sample 24
 DFHZNEP 28, 133
 DIMOPS 26
 Directory
 files, and OPEN or CLOSE commands 51
 Directory Maintenance 17, 40
 DISCONNECT command 44
 description 91
 detaching CICS Connector 23
 Operator Control Task 32
 disconnecting the PDM 23
 DSA *See* Dynamic Storage Area (DSA)
 dump codes 54
 DUMP command
 description 93
 dumping virtual storage areas 59
 Operator Control Task 32
 resolving errors 54
 Dump Control Program (DCP) 61
 Dynamic Storage Area (DSA)
 memory storage 26
- E**
- EIBTASKN field 20
 EIBTRMID field 20
 EIP. *See* Executive Interface Program
 environment
 creating 24
 setup problems, CICS 55
 error messages 46, 54
 file open or close 50
 EUPD. *See* Exclusive Update

Event Control Block (ECB) 30
Exclusive Update (EUPD) 51
execution JCL 71
Executive Interface Program
(EIP) 61
exit program. See user exit

F

File Control Program (FCP) 61
FILE parameter, CSTXOPRM
152
FILENM parameter, CSTXOPRM
153
files
 exception conditions 51
 opening and closing
 dynamically 50
force option on disconnect 24, 48
settings 91

G

global views, RDM 35
GROUP parameter, CSTXOPRM
151
GROUPNM parameter,
CSTXOPRM 151

H

heaps, RDM 35

I

ICOR parameter, CSTXOPRM
140
ICOR status 31, 49, 73
 retries 90, 140
ID
 operator control program 142
 terminal and nonterminal tasks
 20
IFRTRY parameter, CSTXOPRM
140
INFOPRE parameter,
CSTXOPRM 140
initialization Program List Table
68

initializing CICS 43
initializing RDM 97
ISRA information menu 129
ISRB PDM information 132
ISRC PDM file information 130
ISRD PDM file usage information
130
ISRE secondary key information
132
ISRF PDM Connector information
131

J

JCL 71

L

LABELN parameter, CSTXOPRM
153
logging See task logging
logical unit of work 19
lost terminal 76
LST1 76

M

MACLIB
 DFHZNEP 28
macros
 CICS Connector 102
macros, assembling
 C\$TXUNCP 155
 CSTXOPRM 135
MANTIS 17, 66, 67
MANTIS applications 39
MAXPACK parameter,
 CSTXOPRM 141
MBLOC parameter, CSTXOPRM
141
MBSIZE parameter, CSTXOPRM
141
MEME parameter
 additional memory 27
 description 142

- memory
 - allocation
 - dynamic CICS 27
 - blocks, maximum 142
 - determining usage 49
 - ICOR status 31
 - maximum number of
 - extensions 88
 - OPEN and CLOSE files 50
 - requesting additional 26
 - shared connector
 - location 141
 - size 141
 - storage 26
 - XPCC packets 26
 - memory storage 26
 - MESSQID parameter,
 - CSTXOPRM 142
 - MRO environment
 - example 79
 - NEP support 76
 - remote transactions 102
 - multiple tasks 20
 - multitask PDM 20
 - communication with 18
- N**
- NEWCOPY, command 72
 - Node Error Program (NEP) 28,
 - 76
 - PURGE command 52
 - nonconversational transactions
 - 22
 - nonterminal tasks
 - ID 20
 - opening files 51
 - NOTO status 21, 59
- O**
- obsolete user exits 128
 - OCPTRID parameter,
 - CSTXOPRM 142
 - OCPTRTM parameter,
 - CSTXOPRM 143
 - OPEN command
 - description 94
 - Directory files 51
 - dynamically opening files 50
 - exceptions 51
 - file groups 94
 - mode 95
 - Operator Control Task 32
 - OPER ABEND. *See* operator control commands
 - OPER CONNECT command
 - activating 24
 - memory allocation 26
 - overview 24
 - recommended use 46
 - OPER CONNECT. *See* operator control commands
 - OPER DISCONNECT command
 - activating 24
 - overview 24
 - OPER DISCONNECT. *See* operator control commands
 - OPER DUMP. *See* operator control commands
 - operating modes
 - attached 43
 - attached central 44
 - central 41, 42
 - PDM 41
 - operator control commands 83
 - ABEND 84
 - CLOSE 85
 - CONNECT 87
 - DISCONNECT 43, 44, 91
 - DUMP 93
 - OPEN 94
 - PURGE 96
 - RDM 97
 - STATUS 98
 - Operator Control Parameter Table (CSTXOTBL) 31, 32
 - Operator Control Parameter Table Macro (CSTXOPRM) 72, 110, 118, 124
 - operator control program ID 142
 - Operator Control Task 32
 - illustrated 33
 - OPNMODE parameter,
 - CSTXOPRM 153

- P**
- packet size, maximum 141
 - page command, BMS, warning 22
 - PDM
 - abends, recovery from 60
 - connecting and disconnecting 23
 - connecting the CICS connector to 46
 - Connector information 131
 - disconnecting the CICS connector from 47
 - file information 130
 - file usage information 130
 - information menu 129
 - multitask environment 20
 - number of tasks 49
 - operating modes 41
 - task flow 25
 - tasks 28
 - PDM tasks
 - association with CICS tasks 20
 - PDML
 - non-task-logging environment 19
 - retrying 31
 - SINON 28
 - PDML commands, trace ID 148
 - PLT. *See* Program List Table
 - Post Processing User Exit PARM and work area 102
 - post-DML user exit 118
 - Post-DML User Exit PARM and work area 102
 - post-process user exit 113
 - pre-DML user exit 116
 - Pre-DML User Exit PARM and work area 103
 - preprocess user exit 110
 - Pre-Process User Exit PARM and work area 103
 - primary files 85
 - problems
 - CFUL status 73
 - CICS task abend 54, 59
 - resolving 53
 - TFUL status 73
 - processing of operator commands 32
 - Program List Table
 - activating OPER CONNECT 24
 - activating OPER DISCONNECT 24
 - Program List Table (PLT)
 - connecting 46
 - disconnecting 47
 - initialization 68
 - termination 69
 - pseudoconversational transactions 22
 - detaching task 28
 - held records 139
 - trace ID 149
 - PURGE command
 - description 52, 96
 - Operator Control Task 32
 - purging tasks 52
 - PUSHPOP parameter, CSTXOPRM 143
- Q**
- queue ID in DCT table 142
- R**
- RC value. *See* return codes
 - RDM command
 - description 97
 - RDM. *See* Relational Data Manager (RDM)
 - RDML tasks, memory for 26
 - RDO transactions 63
 - RDODIRM 63
 - RDOGNIT 63
 - RDOISRX 63
 - RDOONEW 63
 - RDOOPER 63
 - RDORDM 63
 - RDOSPEC 63
 - recovery, system 60
 - recovery, task 28

Relational Data Manager (RDM)
 command 97
 global views 35
 heaps 35
 initializing 97
 interface with CICS 35
 resident module 35
 stopping 97
 task processing 36
 RESET command 32
 resolving problems 53
 resource allocation 24
 resource allocations
 exceeding 31
 resources, connector 62
 restarting
 in task logging environment 19
 retries
 CFUL status 139
 default values 31
 ICOR status 140
 maximum number 89
 specifying number of 31
 TFUL status 145
 return codes 105, 111, 117, 121
 RSSO processing 20, 30, 32
 run-time statistics 49

S

sample transactions 63
 sealed interface 54
 secondary key information 132
 sequential devices,
 recommendation for
 connecting 46
 shared connector memory
 location 141
 shared connector memory size
 141
 Signed On Table Block (SOTB)
 104
 SIT. See System Initialization
 Table
 source programs 133
 SPECTRA 17
 accessing files 66
 configuring 38
 processing 37

stalled state 92
 statistics
 printing 48
 run-time 49
 status codes 31
 CFUL 49, 73
 ICOR 49
 NOTO 59
 TFUL 49, 73
 STATUS command
 description 98
 determining connection status
 49
 Operator Control Task 32
 stopping RDM 97
 storage allocation 26
 SUPRA Server
 components 34
 SYNC parameter, CSTXOPRM
 144
 synchronization, task 144
 System Initialization Table (SIT)
 70
 system recovery 60

T

TABLE parameter, CSTXOPRM
 139
 task detach user exit 124
 Task Detach User Exit (TDUE)
 30
 Task Detach User EXIT PARM
 and work area 102
 Task Log File 19
 task logging 19
 Task Related User Exit (TRUE)
 illustrated 33
 memory allocation 27
 memory storage 26
 task synchronization user exit
 122

tasks

- abend 29
- association with terminal 20
- CFUL status 31
- component, memory for 26
- concurrent 19
- defined 18, 19
- detach management 24
- detaching 29, 30
- Directory Maintenance 40
- errors 54
- flow to PDM 25
- ID 20, 28
- MANTIS 39
- maximum number 88
- memory 27
- nonterminal 51
 - disconnecting 24
 - ID 20
- number of concurrent 144
- number of PDM 49
- PDM and CICS 20
- pseudoconversational 139
- purging 52
- RDML 26
- recovery 28
- Relational Data Manager (RDM) 36
- SPECTRA 37
- SUPRA Server 19
- synchronization 144
- terminal
 - ID 20
- TASKS parameter, CSTXOPRM 144
- TDUE parameter, CSTXOPRM 145
- TDUE user exit 124
- TDUEABCD 125
- TDUECFRM 125
- TDUEENTRN 125
- TDUEGADR 126
- TDUERSFG 125
- TDUESOTF 125
- TDUESYVF 125
- TDUETCOM 126
- TDUETID 125
- TDUETMSG 125
- Temporary Storage Control Program (TSP) 61

terminal

- activating OPER CONNECT 24
- activating OPER DISCONNECT 24
 - association with tasks 20
- BSAM sequential 24
- ID 28
- Operator Control Task 32
- terminal tasks
 - ID 20
- terminal, lost 76
- termination Program List Table 69
- TFUL parameter, CSTXOPRM 145
- TFUL status 31, 49, 73
 - retries 89, 145
- threads
 - default number 145
 - number of 87
 - TFUL status 31
- THREADS parameter, CSTXOPRM 145
- TOTAL, compatibility 146
- TOTC parameter, CSTXOPRM 146
- TOTINT user exit 128
- TPOP parameter, CSTXOPRM 146
- TPOP user exit 110
- TPOPDML 113
- TPOPGADR 114
- TPOPRDMF DML flag 113
- TPOPTCOM 114
- TPOPTOTF TOTAL COMPATIBILITY flag 113
- TPOS parameter, CSTXOPRM 146
- TPOS user exit 116
- TPOSCFRF 119
- TPOSDML 119
- TPOSGADR 120
- TPOSSYVF 119
- TPOSTCOM 120
- TPRE parameter, CSTXOPRM 147
- TPRE user exit 116
- TPRECFRF 116
- TPREDML 116
- TPREGADR 116

TPRESYVF 116
 TPRETCOM 116
 TPRP parameter, CSTXOPRM
 147
 TPRP user exit 110
 TPRPDML 110
 TPRPDMLF DML flag 110
 TPRPGADR 111
 TPRPSYVF 110
 TPRPTCOM 111
 TPRPTOTF TOTAL
 COMPATIBILITY flag 110
 trace ID
 PDML commands on entry 148
 PDML commands on exit 148
 pseudoconversational tasks
 149
 trace table entries 58
 TRACE1 parameter, CSTXOPRM
 148
 TRACE2 parameter, CSTXOPRM
 148
 TRACE3 parameter, CSTXOPRM
 149
 transaction
 pseudoconversational
 detaching task 28
 transaction dumps 58
 Transaction List Table (XLT) 70
 transactions
 conversational 21
 defined 18
 defining ID 150
 defining to CICS 67
 ISRA 129
 ISRB 132
 ISRC 130
 ISRD 130
 ISRE 132
 ISRF 131
 nonconversational 22
 operator control program ID
 142
 prefix 140
 pseudoconversational 22
 remote 102
 samples 63
 supplied 129
 types 21
 TRUE See Task Related User
 Exit (TRUE)

TSYN parameter, CSTXOPRM
 149
 TSYN user exit 122
 TSYNCOMF 122
 TSYNGADR 123
 TSYNRSEF 122
 TSYNSTPF 122
 TSYNSYVF 122
 TSYNTCOM 123
 TUID exit 104
 TUID parameter, CSTXOPRM
 149
 tuning, retry counters 31

U

user exits
 common user exit PARM and
 work area 102
 CSTXDBXT 128
 CSTXKCCX 128
 CSTXMRXT 99, 127
 CSTXUDAT 99, 107, 109
 name 145, 146, 147, 149
 obsolete 128
 overview 100
 post processing user exit
 PARM and work area 102
 Post-DML User Exit PARM and
 work area 102
 Pre-DML User Exit PARM and
 work area 103
 Pre-Process User Exit PARM
 and work area 103
 task detach user exit and work
 area 102
 TDUE 99, 124, 126
 TOTINT 128
 TPOP 99, 110, 113, 115
 TPOS 99, 116, 118, 121
 TPRE 99, 116, 121
 TPRP 99, 110, 112
 TSYN 99, 122, 123
 TUID 99, 104, 106
 User Synchronization PARM
 and work area 103
 User Task ID Exit PARM and
 work area 103
 XDBINIT 128
 XKCREQ 128
 XPCFTCH 128

User Synchronization PARM and
work area 103
User Task ID Exit PARM and
work area 103
USERTID parameter,
CSTXOPRM 150
UTIDGADR 105
UTIDTCOM 105

V

virtual storage, dumping 59
VSAM files 66
VSE
memory for XPCC packets 26

X

XDBINIT user exit 128
XKCREQ user exit 128
XLT table entry for OPER
transaction code 24
XLT. See Transaction List Table
XPCC packets 26
XPCFTCH user exit 128