

CICS Transaction Server for z/OS



# CICSplex SM Problem Determination

*Version 3 Release 1*



CICS Transaction Server for z/OS



# CICSplex SM Problem Determination

*Version 3 Release 1*

**Note!**

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 157.

**Sixth Edition (July 2010)**

This edition applies to Version 3 Release 1 of CICS Transaction Server for z/OS, program number 5655-M15, and to all subsequent versions, releases, and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

© Copyright IBM Corporation 1994, 2010.

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

---

# Contents

<b>Preface</b> . . . . .	ix
Who this book is for . . . . .	ix
Notes on terminology . . . . .	ix
CICS system connectivity . . . . .	ix
<b>Summary of changes</b> . . . . .	xi
Changes made to this book for CICS Transaction Server for z/OS, Version 3	
Release 1 . . . . .	xi
Changes made to this book for CICS Transaction Server for z/OS, Version 2	
Release 3 . . . . .	xi
Changes to this book for CICS Transaction Server for OS/390 Release 3 . . . . .	xi

---

## Part 1. Approach to problem determination. . . . . 1

<b>Chapter 1. Introduction</b> . . . . .	3
What is problem determination? . . . . .	3
How this book can help you . . . . .	3
<b>Chapter 2. CICSplex SM system overview</b> . . . . .	5
The structure of CICSplex SM . . . . .	5
The Web User Interface . . . . .	6
The ISPF end-user interface . . . . .	7
CMAS networks and registration . . . . .	9
The structure of the CMAS . . . . .	10
The agents in a MAS . . . . .	12
The ESSS and data spaces . . . . .	12
Common components . . . . .	13
Kernel Linkage . . . . .	13
Trace Services . . . . .	13
Message Services. . . . .	13
Common Services. . . . .	14
Data Cache Manager . . . . .	14
Queue Manager . . . . .	14
Data Repository . . . . .	14
Communications . . . . .	15
<b>Chapter 3. Identifying a problem.</b> . . . . .	17
Has CICSplex SM run successfully before? . . . . .	17
Have any changes been made since the last successful run? . . . . .	17
Are there any messages that could explain the problem? . . . . .	18
Does the problem occur at specific times? . . . . .	18
Does the problem affect specific parts of the environment? . . . . .	18
Common types of problem . . . . .	18
<b>Chapter 4. Sources of information</b> . . . . .	21
Your own documentation . . . . .	21
Change log . . . . .	21
Manuals . . . . .	21
Online diagnostic aids . . . . .	21
Messages. . . . .	22
Symptom strings . . . . .	22
LOGREC records . . . . .	22
Traces . . . . .	23

<b>Part 2. Tools for problem determination . . . . .</b>	<b>25</b>
<b>Chapter 5. Using trace . . . . .</b>	<b>27</b>
The CMAS and MAS trace facilities . . . . .	27
Tracing in a CMAS . . . . .	27
Tracing in a MAS . . . . .	27
Types and levels of tracing . . . . .	28
Controlling the amount of tracing in a CMAS or MAS . . . . .	29
Using the end-user interface to control tracing . . . . .	29
Interpreting CMAS and MAS trace entries . . . . .	30
Formatting CMAS and MAS trace entries . . . . .	31
Trace formatting options on the host . . . . .	31
Trace formatting JCL. . . . .	33
The CAS LU 6.2 communications trace . . . . .	34
Tracing LU 6.2 communications. . . . .	34
Interpreting the LU 6.2 communications trace. . . . .	35
Web User Interface trace services . . . . .	36
Setting trace flags using the WUITRACE parameter . . . . .	36
Setting trace flags through COVC . . . . .	36
The available trace flags . . . . .	37
Exception traces . . . . .	38
<b>Chapter 6. Using dumps . . . . .</b>	<b>39</b>
Unexpected dumps . . . . .	39
CICSplex SM dumps under MVS . . . . .	39
CICSplex SM dumps under CICS. . . . .	40
CICSplex SM-requested dumps . . . . .	42
Abends in the CAS . . . . .	42
CMAS initialization failures . . . . .	42
MAS initialization failures . . . . .	42
ESSS program call (PC) routine failures . . . . .	43
User-requested dumps . . . . .	44
Using the MVS DUMP command . . . . .	44
Using the ISPF end-user interface . . . . .	45
Managing dumps in MVS . . . . .	46
Using dumps with the Web User Interface . . . . .	47
The available dump codes. . . . .	47
<b>Chapter 7. Displaying and formatting dumps with IPCS . . . . .</b>	<b>49</b>
Using the CICSplex SM dump formatting routine . . . . .	49
Formatting a CICSplex SM SDUMP . . . . .	49
<b>Chapter 8. Using the ESSS utility (EYU9XEUT) . . . . .</b>	<b>53</b>
The EYU9XEUT options . . . . .	53
Dumping data structures (DUMP) . . . . .	53
Reloading broadcast functions (RELOAD) . . . . .	54
Stopping the ESSS (TERMINATE). . . . .	54
The EYU9XEUT JCL. . . . .	54
Using the ESSS Information Display Utility (EYU9XENF) . . . . .	55
<b>Chapter 9. Using the online utility transaction (COLU) . . . . .</b>	<b>57</b>
The COLU transaction . . . . .	57
Valid keywords for component CHE . . . . .	57
Valid keywords for component COM . . . . .	58
Valid keywords for component KNL . . . . .	58
Valid keywords for component QUE . . . . .	59

Valid keywords for component SRV . . . . .	60
Valid keywords for component TOP . . . . .	60

<b>Chapter 10. Using the interactive debugging transactions (COD0 and CODB) . . . . .</b>	<b>61</b>
Running the debugging transactions . . . . .	61
Method-level debugging with COD0 . . . . .	62
Issuing commands recursively . . . . .	62
Issuing commands that alter CICSplex SM . . . . .	62
ALLOC (allocating a resource) . . . . .	63
ATTACH (attaching a method) . . . . .	64
CALL (calling external CICS programs and transactions) . . . . .	66
CAPTURE (capturing and printing a view) . . . . .	67
DUMP (displaying and altering data) . . . . .	68
EXEC (executing a method) . . . . .	69
EXIT (exiting COD0) . . . . .	70
HELP (getting online help) . . . . .	70
LIST (listing tasks and allocated resources) . . . . .	70
POST (posting an ECB) . . . . .	82
PRINT (printing data areas under CICS/ESA, CICS TS) . . . . .	82
PURGE (purging an allocated resource) . . . . .	83
START (starting a method in the CMAS) . . . . .	84
TRACE (setting CICS and CICSplex SM trace flags) . . . . .	84
TRACK (setting trace flags by calling structure) . . . . .	85
TRAP (setting trace flags for a method) . . . . .	86
Displaying a MAL from COD0 . . . . .	86
System-level debugging with CODB . . . . .	90
CODB commands . . . . .	91
The COMP ID field . . . . .	92
The ADDR field . . . . .	92
The ALET field . . . . .	92
The PF Key prompts . . . . .	92
The MSG field . . . . .	94
The memory display area . . . . .	94
CODB altering memory . . . . .	94
Accessing CODB from COD0 . . . . .	95

<b>Chapter 11. Using PlexManager diagnostic facilities . . . . .</b>	<b>97</b>
The DIAGxxxx views . . . . .	97
DIAGMSG (Diagnostic facilities) . . . . .	97
DIAGSESS (Connected sessions) . . . . .	98
DIAGSYS (Connected CASs) . . . . .	100
Extended diagnostic mode . . . . .	101
Status messages . . . . .	101
Internal tracing . . . . .	101
Disabling error recovery . . . . .	102

---

**Part 3. Investigating and documenting a problem. . . . . 103**

<b>Chapter 12. Abends, stalls, and bottlenecks. . . . .</b>	<b>105</b>
Investigating abends . . . . .	105
MVS abends . . . . .	105
CICS abends . . . . .	105
Investigating stalls . . . . .	106
An undetermined stall condition . . . . .	106
A suspected loop . . . . .	107

A suspected wait . . . . .	107
Investigating bottlenecks . . . . .	107
<b>Chapter 13. Investigating output and system management problems</b>	<b>109</b>
Incomplete operations data returned . . . . .	109
No entry in the MAS view . . . . .	109
INACTIVE status . . . . .	110
ACTIVE status . . . . .	112
Missing monitor data . . . . .	112
Unexpected real-time analysis results . . . . .	114
An example SAM problem . . . . .	114
An example MRM problem . . . . .	115
Unexpected workload management routing decision . . . . .	116
Is dynamic routing enabled? . . . . .	117
Which workload is active? . . . . .	117
Is the workload being separated? . . . . .	118
Application programming interface problems . . . . .	120
<b>Chapter 14. Investigating Web User Interface problems</b>	<b>121</b>
Server and browser messages . . . . .	121
Server messages . . . . .	121
Browser messages . . . . .	121
COVC status panel . . . . .	121
COVC debugging commands . . . . .	122
Running the COVC transaction for debugging . . . . .	122
Typical end-user problems . . . . .	123

---

## Part 4. Working with IBM to solve a problem . . . . . 125

<b>Chapter 15. IBM Program Support</b>	<b>127</b>
When to contact the Support Center . . . . .	127
Working with the Support Center . . . . .	127
What the Support Center needs to know . . . . .	128
What happens next . . . . .	129
<b>Chapter 16. APARs, fixes, and PTFs</b>	<b>131</b>
The APAR process . . . . .	131
Collecting the documentation for an APAR . . . . .	131
Documentation needed for problems with CICSplex SM . . . . .	132
Sending the documentation to IBM . . . . .	132
Applying the fix . . . . .	133
The APAR becomes a PTF . . . . .	133

---

## Part 5. Appendixes . . . . . 135

<b>Appendix A. CICSplex SM naming standards</b>	<b>137</b>
The format of names . . . . .	137
Element type identifiers . . . . .	137
Component identifiers . . . . .	138
Major components of CICSplex SM . . . . .	139
<b>Appendix B. System parameters for problem determination</b>	<b>141</b>
Specifying system parameters . . . . .	141
The problem determination parameters . . . . .	142

<b>Bibliography</b> . . . . .	145
The CICS Transaction Server for z/OS library . . . . .	145
The entitlement set . . . . .	145
PDF-only books . . . . .	145
Other CICS books . . . . .	147
Determining if a publication is current . . . . .	147
<b>Accessibility</b> . . . . .	149
<b>Index</b> . . . . .	151
<b>Notices</b> . . . . .	157
Programming interface information . . . . .	158
Trademarks. . . . .	158
<b>Sending your comments to IBM</b> . . . . .	159



---

## Preface

This book is intended to help you determine the cause of problems in a system running CICSplex<sup>®</sup> System Manager for CICS<sup>®</sup> Transaction Server for z/OS<sup>®</sup>. It contains a structural overview of the CICSplex SM system, guidance for investigating and documenting CICSplex SM problems, and instructions for working with the IBM<sup>®</sup> Support Center and submitting Authorized Program Analysis Reports (APARs).

---

## Who this book is for

This book is for system programmers who are responsible for diagnosing CICSplex SM systems. You are assumed to have a good knowledge of CICS and CICSplex SM. You also need to be familiar with the books that tell you how to set up and use CICSplex SM.

---

## Notes on terminology

In the text of this book, the term **CICSplex SM** (spelled with an uppercase letter *P*) means the CICSplex SM element of CICS TS for z/OS. The term **CICSplex** (spelled with a lowercase letter *p*) means the largest set of CICS systems to be managed by CICSplex SM as a single entity.

Other terms used in this book are:

**KB** 1 024 bytes

**MB** 1 048 576 bytes

**MVS™** MVS/Enterprise Systems Architecture SP

---

## CICS system connectivity

This release of CICSplex SM can be used to control CICS systems that are directly connected to it.

For this release of CICSplex SM, the connectable CICS systems are:

- CICS Transaction Server for z/OS 3.1
- CICS Transaction Server for z/OS 2.3
- CICS Transaction Server for z/OS 2.2
- CICS Transaction Server for OS/390<sup>®</sup> 1.3

You can use this release of CICSplex SM to control systems running supported releases of CICS that are connected to, and managed by, your previous release of CICSplex SM. However, if you have any directly-connectable release levels of CICS, as listed above, that are connected to a previous release of CICSplex SM, you are strongly recommended to migrate them to the current release of CICSplex SM, to take full advantage of the enhanced management services. See the *CICS Transaction Server for z/OS Migration from CICS TS Version 2.3* for information on how to do this.

Table 1 on page x shows which supported CICS systems can be directly connected to which releases of CICSplex SM.

| *Table 1. Directly-connectable CICS systems by CICSplex SM release*

CICS system	CICSplex SM component of CICS TS 3.1	CICSplex SM component of CICS TS 2.3	CICSplex SM component of CICS TS 2.2	CICSplex SM component of CICS TS 1.3
CICS TS 3.1	Yes	No	No	No
CICS TS 2.3	Yes	Yes	No	No
CICS TS 2.2	Yes	Yes	Yes	No
CICS TS 1.3	Yes	Yes	Yes	Yes
TXSeries 4.3.0.4	No	Yes	Yes	No
TXSeries 5.0	No	Yes	Yes	No

---

## Summary of changes

This book is based on the CICS Transaction Server for z/OS, Version 2 Release 3 edition of *CICSplex System Manager Problem Determination*. It has been updated to incorporate changes made for CICS Transaction Server for z/OS, Version 3 Release 1.

Changes made since the last edition are marked by vertical bars in the left margin.

---

### Changes made to this book for CICS Transaction Server for z/OS, Version 3 Release 1

CICSplex SM support for the CICS for Windows component of IBM TXSeries (also known as Windows NT 4.3 and Windows NT 5.0) is no longer provided in CICS Transaction Server for z/OS, Version 3 Release 1. Therefore, it is not now possible to set up a CICSplex SM remote MAS agent for Windows.

However, you can continue to use the CICS Transaction Server Version 2.3 or Version 2.2 for CICSplex SM support of TXSeries.

---

### Changes made to this book for CICS Transaction Server for z/OS, Version 2 Release 3

The more significant changes for this edition are:

- Sections have been added to facilitate problem determination in a remote MAS.
- “The problem determination parameters” on page 142 includes several new parameters including ESDUMPCOM and EYUTRACE.

There has been a change in CICSplex SM field naming conventions in this release. Data set name fields such as DSNAME, and file name fields such as LOCFILE and REMFILE are now case-sensitive. When entering data set and file names into the CICSplex SM interfaces (EUI, API and WUI), ensure that you enter the data in the correct case. In previous releases of CICSplex SM, the data set names and file names are automatically converted to upper case.

---

### Changes to this book for CICS Transaction Server for OS/390 Release 3

This book has been updated to reflect the new and changed functions of CICSplex SM for CICS Transaction Server for OS/390 Release 3. For example:

The CICSplex SM dynamic routing capability has been extended to include:

- EXEC CICS START commands that are associated with a terminal
- Distributed program link (DPL) requests:
  - The CICS Web interface
  - The CICS Gateway for Java™
  - EXCI calls
  - CICS Client ECI calls
  - Distributed Computing Environment (DCE) remote procedure calls (RPCs)
  - Open Network Computing (ONC) remote procedure calls (RPCs)
  - Any function that issues an EXEC CICS LINK PROGRAM request
  - Internet Inter Object Request Block Protocol (IIOP)

## summary of changes

- CICS business transaction services processes and activities

A new component, BAS, has been added to the trace components.

Two new system parameters, BASTRACE and BASCONMSG, have been added for problem determination.

---

## **Part 1. Approach to problem determination**

The first part of this book provides an introduction to the problem determination process and an overview of the CICSplex SM system. It introduces the facilities for checking out the CICSplex SM system and will help you to identify the type of problem you are having.



---

## Chapter 1. Introduction

This book is designed to help you find the causes of problems with your CICSplex SM system.

### Important

Keep in mind that CICSplex SM is a tool for managing the CICS systems at your enterprise. As you investigate a potential problem in your CICS environment, be sure to distinguish between problems in managing your CICS systems and problems with the CICS systems themselves.

---

### What is problem determination?

Usually, when you are investigating a problem, you start with a symptom, or set of symptoms, and try to trace them back to their cause. This process is called *problem determination*, and it is important to realize that it is not the same as problem solving.

Often, the process of problem determination enables you to solve the problem. For example:

- If you find that the cause of a problem is conflicting CICSplex SM topology definitions, you can solve the problem by correcting the definitions.
- If you find that the cause of a problem is within CICS, you can solve the problem by modifying CICS. (For example, if CICSplex SM's Workload Manager will not route to a target region because there is no CICS connection between the routing region and the target region, you can create the links between the systems.)

However, you may not always be able to solve a problem yourself after determining its cause. For example:

- An unexpected message may be caused by an unexpected response from another product.
- If you think the cause of a problem is in the CICSplex SM code, you need to contact your IBM Support Center for assistance.

---

### How this book can help you

In this book, we start with the symptoms of a problem, and try to help you use those symptoms to classify it. For each type of problem, we suggest possible causes, and techniques you can use to determine the actual cause.

You should always assume first that the problem has a simple cause, such as a definition error. If, as a result of investigation, you find that the cause of the problem is not straightforward, then consider possible causes that may be more difficult to identify. If further investigation still does not provide an answer, it is possible that the cause of the problem is in the CICSplex SM code itself. If this appears to be the case, you need to contact your IBM Support Center.

Figure 1 on page 4 will help you decide which part of the book to read first.

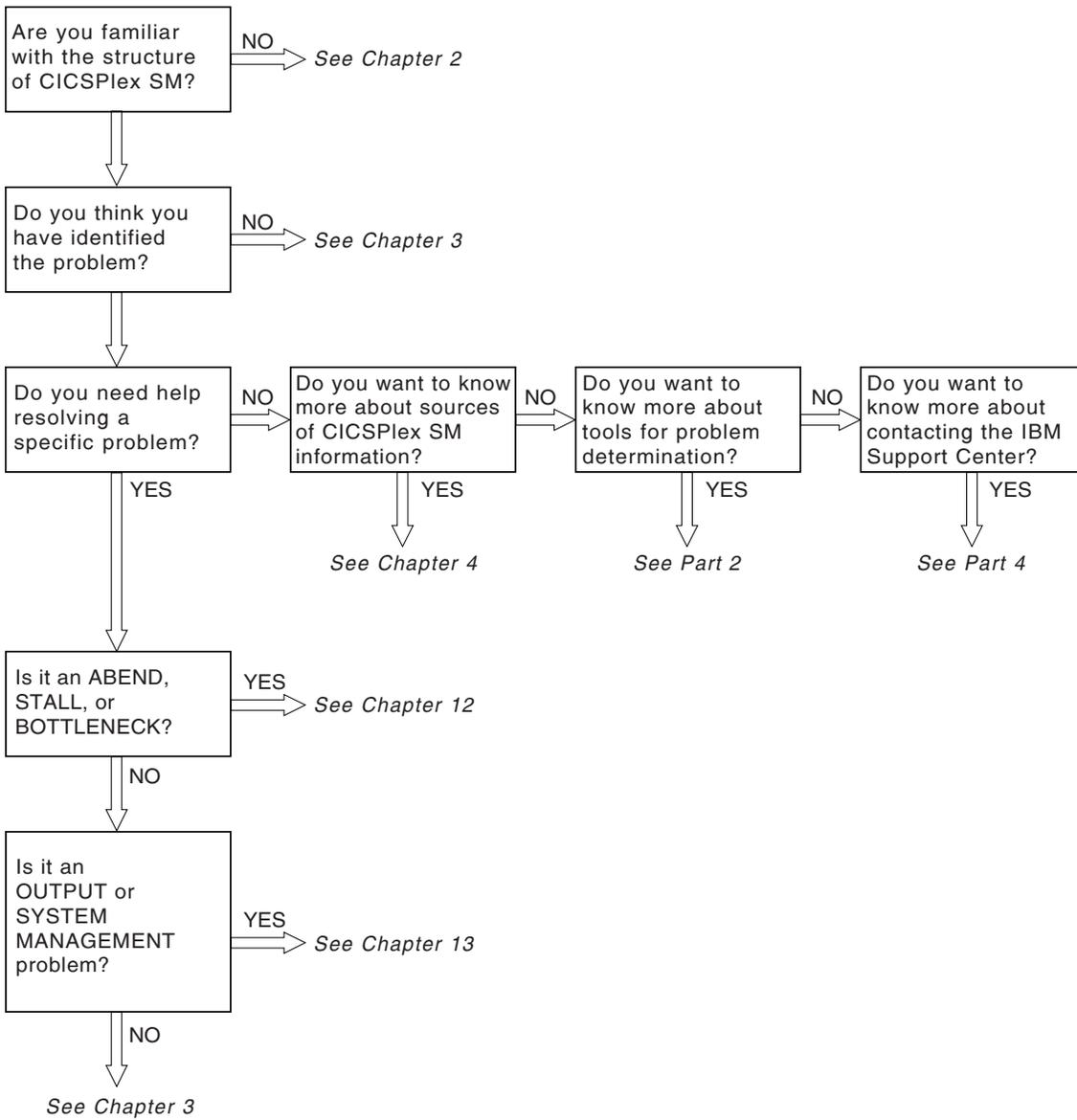


Figure 1. Where to look first

---

## Chapter 2. CICSplex SM system overview

This chapter provides an overview of CICSplex SM processing. It introduces the components of CICSplex SM, and describes how they work together to provide effective management of your CICS systems.

---

### The structure of CICSplex SM

CICSplex SM makes use of a distributed system management architecture that is based on a manager-and-agent model. In CICSplex SM, the agent runs in a managed CICS system, otherwise known as a managed application system (MAS). The agent is in constant communication with a manager, called a CICSplex SM address space (CMAS). This communication allows the manager to monitor and control the CICS system. The manager consolidates data from, and distributes actions to, the individual agents. The manager is also responsible for basic management applications, such as resource monitoring and workload management.

A typical CICSplex configuration would consist of many agents under the control of a single manager. In a more complex environment, there might be multiple managers, each controlling multiple agents. In order to achieve the distributed system management goal of a single-system image, these managers are normally connected to each other.

Another important aspect of distributed system management that is provided by CICSplex SM is operation from a single point of control. In CICSplex SM, the single point of control is either:

- The Web User Interface (WUI). The WUI runs on a standard web browser using TCP/IP to contact a Web User Interface server running on a dedicated CICS Transaction Server for z/OS, Version 3 Release 1 CICSplex SM local MAS connected to a CICS Transaction Server for z/OS, Version 3 Release 1 CMAS.
- An ISPF end-user interface (EUI). The EUI is provided by a component called the InfoManager. The InfoManager, which runs in a user's TSO session, must have a means of sending requests to and receiving responses from one or more CMASs. The coordinating address space (CAS) is responsible for establishing connections between the InfoManager and CMASs. A CAS normally resides in each MVS image where a CMAS is run. Just as CMASs are normally connected, so are CASs. In addition, if there is an MVS image where no CMAS is run but where a TSO session expects to use the end-user interface, a CAS must reside on that MVS image.

Providing the single-system image and single point of control is a responsibility shared by many parts of the CICSplex SM system. An end user, from a TSO session, enters the ISPF interface via the InfoManager. The InfoManager calls on the CAS to locate a CMAS capable of processing the user's request. The InfoManager then communicates with the target CMAS, using links between CASs, if necessary. So it is the CAS that provides the single point of control by ensuring that the InfoManager is able to communicate with whatever CMAS is needed to perform a given request. But the CMAS distributes the request to multiple CMASs and MASs, as appropriate, and then consolidates the results of the request for presentation to the user via the InfoManager. So it is the CMAS that provides the single-system image by knowing which CMASs and MASs must participate in the processing of a request, routing the request to those destinations, and consolidating the results.

## the structure of CICSPlex SM

In addition to the visible parts of the CICSPlex SM system that either manifest themselves as MVS address spaces (such as the CAS) or run within existing address spaces (such as the agent code for a managed CICS system, which runs in the CICS address space), there is one largely invisible part of the system that is also an address space: Environment Services System Services (ESSS). An ESSS address space resides in each MVS image where a CMAS is run. The ESSS is automatically created when the first CMAS is started in a given MVS image and it remains for the life of the IPL. The ESSS provides the cross-memory services used for communication between a manager and agents when they reside on the same MVS image. It also serves as the owner of all data spaces used by the product, which enables data spaces that are shared between a CMAS and a MAS to survive the shutdown of either.

Figure 2 illustrates the basic structure of the CICSPlex SM system.

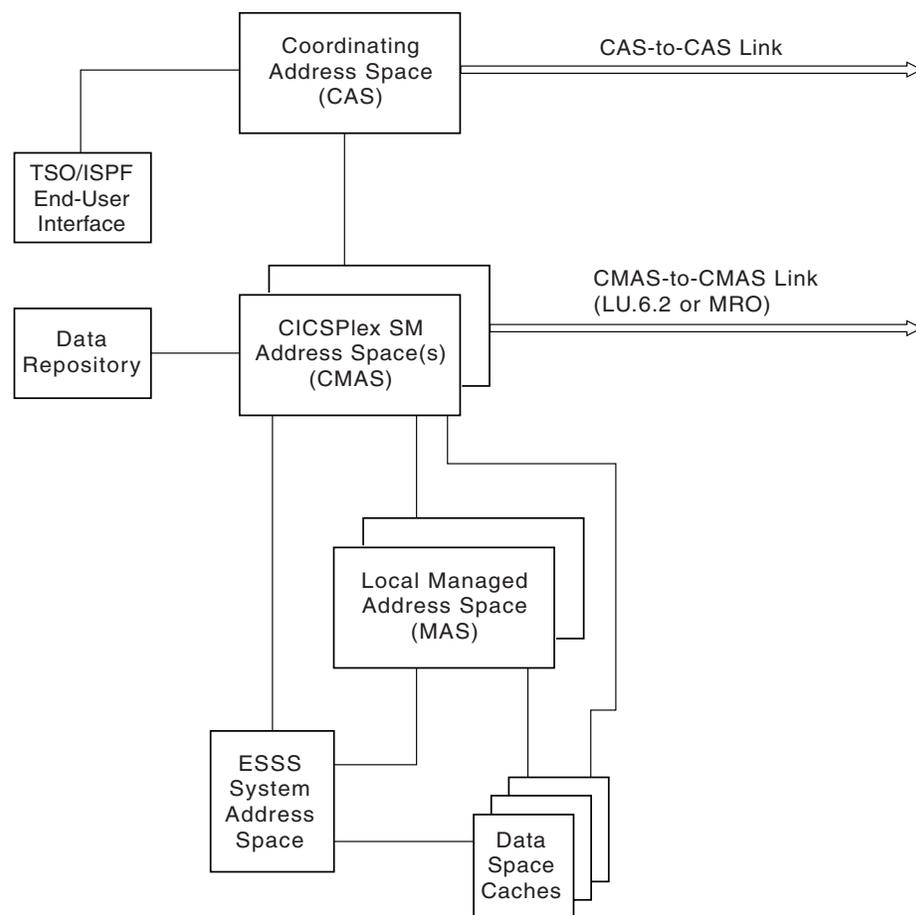


Figure 2. The CICSPlex SM system

---

## The Web User Interface

The CICSPlex SM Web User Interface is an easy-to-use interface that you can use to carry out the operational and administrative tasks necessary to monitor and control CICS resources. You can link to the Web User Interface from any location that can launch a web browser.

The Web User Interface is accessed using standard web browser software in contact with a dedicated CICS region acting as a Web User Interface server. You

can have more than one Web User Interface server active; for example, you may have a requirement for different languages to be used or different systems available to different servers. The web browser client contacts the web server by an HTTP request via the CICS Web Interface.

The Web User Interface server runs as a CICSplex SM local MAS and communicates with the managed resources via the CMAS to which it is connected. This CMAS needs to manage all CICSplexes that the WUI server needs to access. This is because the WUI server acts as a CICSplex SM API application. However, it is not necessary for the CMAS, to which the WUI connects, to be managing any of the MASs in these CICSplexes.

All the menu and view definitions are stored on a server repository. There is one repository for each Web User Interface server. The menu and view definitions can be exported for backup purposes, for distributing definitions to other servers, and for migrating menus and views between product releases.

The Web User Interface provides an administration transaction COVC that allows you to:

- Start and shutdown the Web User Interface server
- View the server status
- View current user activity and terminate active user sessions
- Import and export repository definitions
- Set trace flags

Because a WUI server is defined as a MAS, it can be monitored using standard CICSplex SM monitoring.

---

## The ISPF end-user interface

When a user invokes the ISPF end-user interface, the InfoManager component is in control of the terminal session. The InfoManager provides presentation services that deliver a multiwindow environment to the end user. To display a CICSplex SM view in a window, the user must specify a context for the request. The context is the name of either a CMAS or a CICSplex from which information is desired.

The InfoManager uses the CAS and two of its components, the LUManager and the DataManager, to establish a communication path from the TSO session to the CMAS that will handle the request. The LUManager provides a generalized communication facility between any two points in communication with a CAS. The DataManager passes data between tasks running in the TSO session and in the CMAS and schedules the running of programs that support request processing. The CAS determines which CMAS is to service requests for a given context. Instances of the DataManager are then established in the TSO address space where the InfoManager is running and in the target CMAS; the two are logically connected via the LUManager. An LUManager session must be a single point-to-point connection. This means that there must be a direct link between any two CASs where such a session is to be established. For full connectivity, there must be a direct link from each CAS to every other CAS.

Figure 3 on page 8 shows a sample CAS network.

## ISPF end-user interface

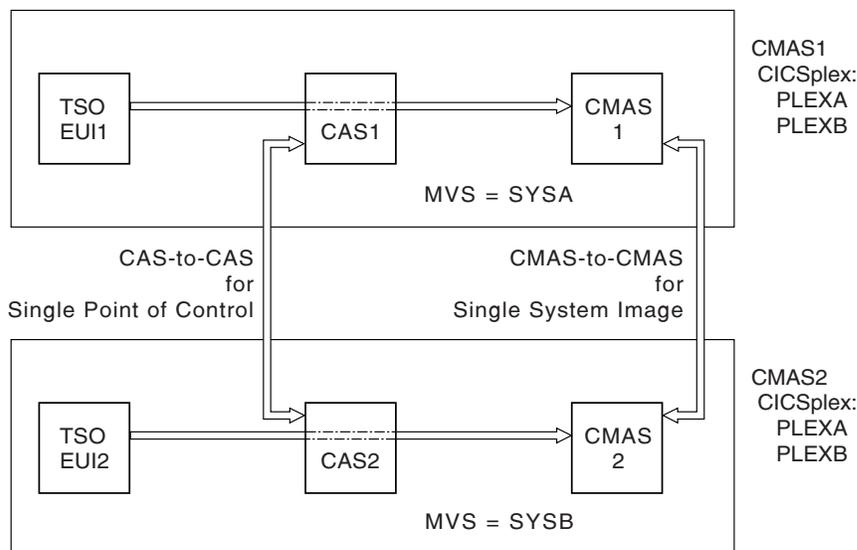


Figure 3. A sample CAS network

When a particular CICSplex SM view is requested, a program is run that supplies data to populate the view. This program is called a selector. A selector runs in a CMAS under an OS TCB that was created as part of the DataManager instance. Note that there may be multiple instances of the DataManager and multiple OS TCBs under which selectors are run, since one is created for each user window that is requesting services. A selector returns records of data to the DataManager, each of which represents a single row in the resulting view. If, for example, a CICSRCN view is run with a context that results in data for three CICS regions, the selector returns three records of data to the DataManager. The selector is rerun whenever a refresh of the data in the view is requested.

Because a CMAS is a special CICS/ESA address space running the CICSplex SM application, the selector must invoke services running under the TCB of the CICS/ESA AP domain. A structure exists within a CMAS to cause the dynamic pairing of the OS task under which the selector runs with a CICS task that can invoke other CICSplex SM routines that must run within the CMAS. WAIT/POST logic is used, together with shared data areas, to direct the invocation of services and the exchange of results between the OS and CICS tasks.

Once a view has been invoked and a selector has supplied the initial data, the user can request an action to be performed against the data. Each action causes a program to run in the CMAS to service the request. This program is called a back-end program. A back-end program is invoked to process an action against a single instance of a resource (such as a transaction in a CICS system). If the user requested that an action be applied to multiple resources, the back-end program is invoked multiple times, once for each resource.

For simple actions, the back-end program processes the action and returns a response indicating the result. For actions that require either more information or confirmation from the user, the back-end program uses a service of the DataManager to run a program called a front-end within the TSO address space. The front-end program presents an ISPF panel to the user and waits for a response. The ISPF panel takes over the screen and all InfoManager windows are temporarily overlaid. The front-end program performs some validation of the data

entered and passes it to the back-end program by way of the DataManager. When processing of the ISPF panel is complete, the InfoManager windows are redisplayed.

Figure 4 illustrates CICSplex SM end-user interface processing.

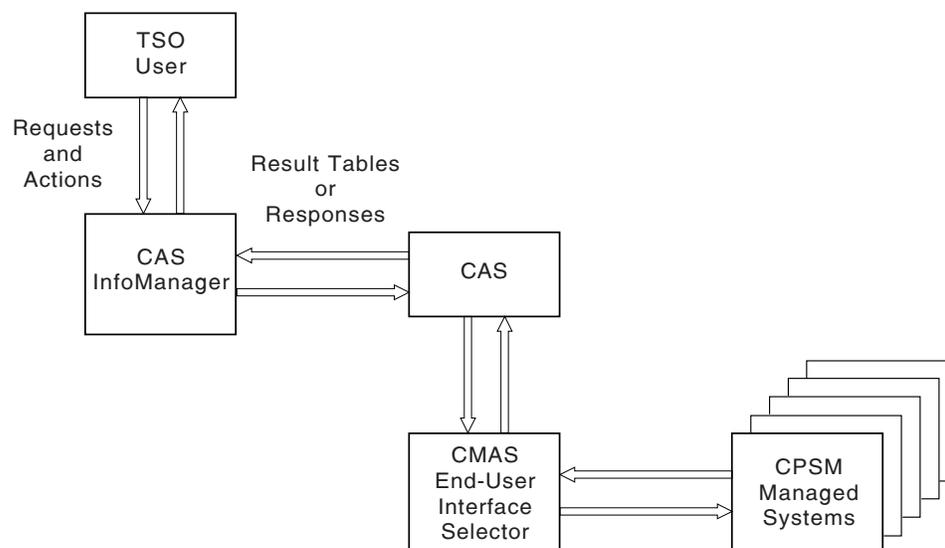


Figure 4. CICSplex SM end-user interface processing

---

## CMAS networks and registration

If more than one CMAS is involved in managing a CICSplex they must all be able to communicate with each other in order to implement single-system image. This communication is also required to allow proper distribution of CICSplex SM definitions from the maintenance point CMAS to other CMASs and to maintain the dynamic CICSplex topology. The maintenance point CMAS is responsible for maintaining the CICSplex definitions in the data repository as well as distributing them to other CMASs.

Unlike the CASs in a network, however, CMASs need not be fully interconnected. The CICSplex SM communications component can deliver a request for remote processing even if the target is not directly connected to the CMAS or MAS where the request originates. The minimum requirement is that you can get from every CMAS to every other CMAS in the network via some route of CMAS-to-CMAS links, no matter how complex. Of course, performance may suffer if excessive transit nodes (those CMASs through which a request must pass on its way to the desired destination) are involved in a request. As a result, more than the minimum number of required communication links are often installed.

Because each CMAS can participate in the management of one or more CICSplexes, it is important for the CAS to know what CICSplexes a CMAS can process requests for. A CMAS provides this information to the CAS during its initialization. The CMAS registers its name and the name of each of the CICSplexes it can manage. The CMAS can also inform the CAS of any changes in the CICSplexes it is managing while it is running. The CAS calls these registered names service points.

## CMAS networks and registration

The CMAS provides the same information about its CICSplexes to the ESSS address space that is running in its MVS image. This is necessary because ESSS establishes the connection between a CMAS and its local MASs. This connection is normally established when the MAS provides its name and the name of the CICSplex it is a member of. So ESSS must be able to find a CMAS that manages the CICSplex named by a MAS.

Figure 5 shows a sample CMAS network and the service points that result.

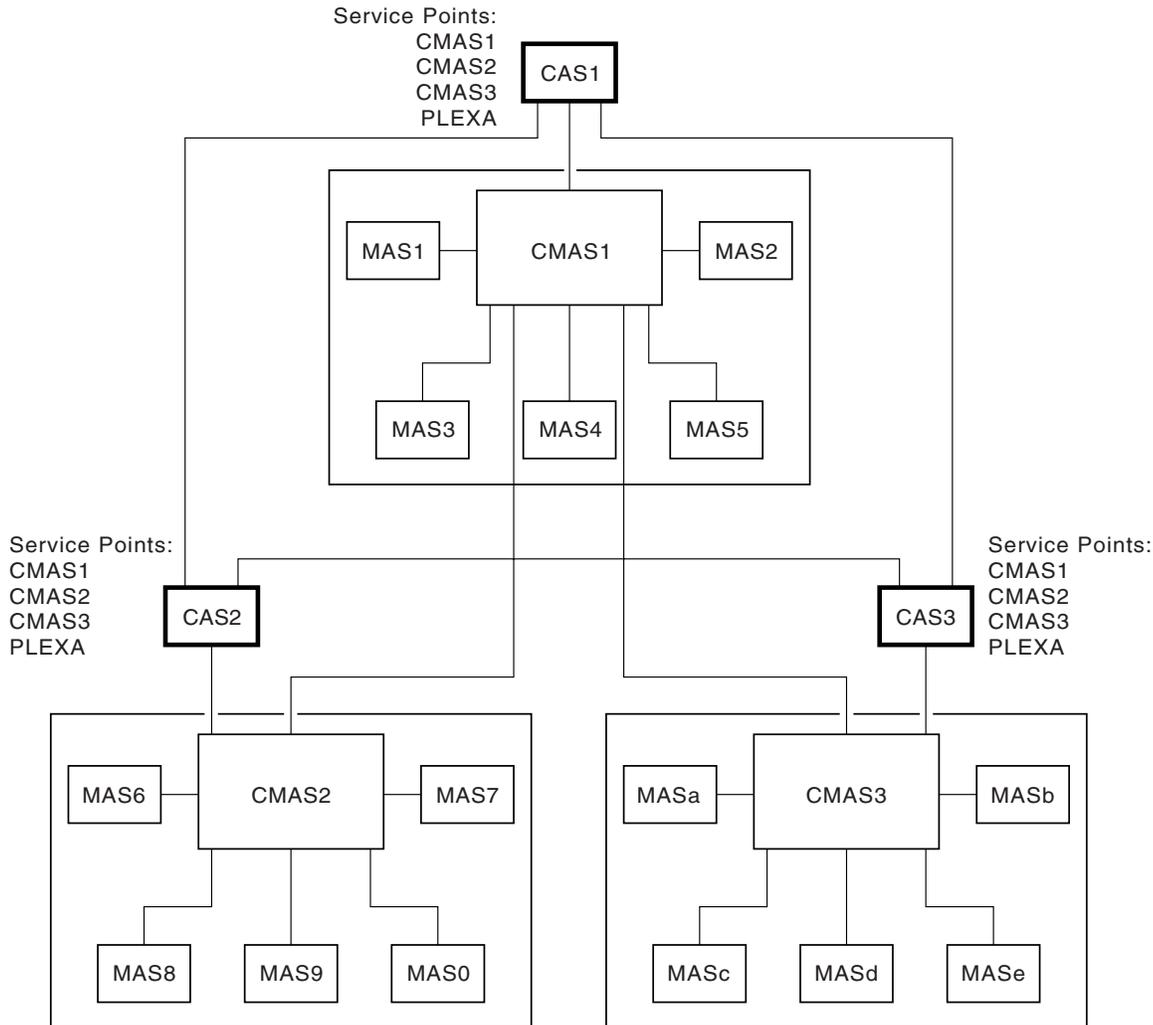


Figure 5. A sample CMAS network

---

## The structure of the CMAS

The CMAS is a special type of CICS/ESA system. To all of the tasks that normally run in CICS/ESA, the CMAS adds a timing services task. A CMAS is started by running program EYU9XECS. This program is responsible for locating the CICSplex SM subsystem and identifying the address space as a starting CMAS. If this is the first CMAS to start after an MVS IPL, EYU9XECS starts a CICSplex SM subsystem. The program then transfers control to program EYU9XSTC, which is the

timing services control program. After initializing, EYU9XSTC attaches DFHSIP, which is the CICS/ESA system initialization program. This starts the CICS/ESA system that runs within a CMAS.

# It is possible to invoke CICSplex SM code automatically during CICS initialization to  
# initialize the region as a CMAS. The preferred method of initializing the region as a  
# CMAS is to use the CPSMCONN=CMAS SIT parameter as described in the  
# CPSMCONN system initialization parameter topic in the CICS System Definition  
# Guide.

# An alternative method is to specify programs to be run in the initialization program  
# list table (PLTPI). For a CMAS, the PLTPI specifies program EYU9XLCS, which  
# issues a CICS START command to start transaction XLEV. This transaction is  
# queued to start when CICS initialization is complete.

| The XLEV transaction runs program EYU9XLEV, which is responsible for creating  
| the run-time environment for a CMAS. The component called Kernel Linkage is  
| responsible for building data structures and controlling the interfaces between other  
| CICSplex SM components. Program EYU9XLEV starts the Kernel Linkage  
| processing. Note that this program is used to create not only the CMAS run-time  
| environment, but also the environment for agent code in local MASs.

The environment that Kernel Linkage creates is called the method call environment. Each program in the CICSplex SM system is called a method. The methods in a single component are grouped together and referred to as a major object, which is really just another name for a component. When one method calls another method, it uses the Kernel Linkage method call services and passes parameters using a data structure called a message argument list (MAL). For each major object, there exist two primary control structures. The first, the major object descriptor block (MODB), is built by Kernel Linkage during initialization and, among other things, contains a directory of all the methods (or programs) that make up the component. The second, the major object environment block (MOEB) is pointed to from the MODB. The MOEB is created during the initialization of each component. While the format of all MODBs is the same, the MOEB for each component is unique and serves to store critical information and to anchor data used by the component.

Once the method call environment has been built, each component that requires initialization is given control to do so. Some components are merely callable services, however others are active parts of the system. Those that are active components (such as Communications, Monitor Services, real-time analysis, and Workload Manager) make calls to Kernel Linkage during initialization to start one or more tasks in the CMAS. These calls identify the method to be run. Kernel Linkage uses the method name and the MODB to determine the proper transaction ID to be used on the EXEC CICS START command. All such transactions have EYU9XLOP defined as their first program. The tasks that are started run as CICS tasks under control of the CICS/ESA system that runs within the CMAS. Each task must establish a unique run-time environment to support method processing. This environment, which runs separate from and in parallel to the other tasks, is created by a program called EYU9XLOP. This program establishes a unique copy of the environment, called an object process, and then actually calls the first method to be run.

After EYU9XLEV has completed the process of sequencing CMAS initialization, it enters a wait state. This wait state is broken only when EYU9XLEV must perform service functions (such as start additional CICS service tasks for the single-system image interface) or when CMAS termination is requested.

---

### The agents in a MAS

For a CICS system to be managed by CICSplex SM, agent code must exist and be in communication with a CMAS. The agent code in a MAS is started in much the same way that CICSplex SM code is started in a CMAS. A program is added to the CICS PLTPI that does a CICS START of a transaction; that transaction invokes EYU9XLEV, the same program that is used in a CMAS.

Once the MAS environment is initialized, a long running task is started that waits for requests from the controlling CMAS. Depending on the type of request received, a method call is made to process the request either synchronously or asynchronously. The long running task is also responsible for starting and stopping the other tasks involved in agent processing, such as monitoring tasks.

Another agent task is responsible for sending a heartbeat to the controlling CMAS. The heartbeat is used to let a CMAS know that the MAS is still able to communicate and to send required data on a regular basis. This data includes a current task count and the health status of the MAS.

The agent code in a managed CICS system is part of the CICSplex SM component called the Managed Application System. This component has an identifier of MAS and its module names have the character N in the fifth position (for example, EYU0NLRT). So MAS is the identifier for both a Managed Application System (a CICS system in which CICSplex SM agent code resides), and for the component that implements the bulk of that agent code.

---

### The ESSS and data spaces

The Environment Services System Services (ESSS) address space is created when the CICSplex SM subsystem is created by the first CMAS started after an MVS IPL. The ESSS is, in MVS terms, a limited function system address space. Once it is started, it never terminates, but neither does it run. The ESSS serves as an anchor point for the data required to establish the connection between a CMAS and its local MASs. It also serves as the owner of all CICSplex SM data spaces and cross-memory services resources. The data in the ESSS private area is updated by program call routines provided by ESSS itself. Since the ESSS does not run after initialization, it is very reliable. This reliability helps to ensure that the cross-memory resources and data spaces remain available until CICSplex SM explicitly deletes them.

CICSplex SM uses MVS data spaces to store some of its data structures because of the potentially large amount of data involved in managing a CICSplex environment. The size of some data structures is directly related to the number of managed systems, while the size of others is related to the number of interconnected CMASs or the system management options in use (such as real-time analysis or monitoring).

Individual data caches are used by each component that has significant storage requirements. Each logical cache can span more than one data space, but no two caches ever share a single data space. So even a simple CICSplex configuration can cause the allocation of many data spaces. While many data spaces may be created, however, CICSplex SM uses only as much storage as is actually required for a given configuration.

---

## Common components

In a system as complex as CICSplex SM, it makes sense to have a foundation of common components on which to build so that common functions can be provided by a single component. CICSplex SM has many such building blocks that are used not only by all who require the service within a CMAS, but also within a MAS when the same services are required.

For a complete list of the major components of CICSplex SM, see “Major components of CICSplex SM” on page 139.

## Kernel Linkage

The role that Kernel Linkage plays in the transfer of control between methods was described in “The structure of the CMAS” on page 10. Kernel Linkage also has several subcomponents that provide services related to maintenance of the basic CICSplex SM environment:

- **Status Services**  
Controls the synchronization between components and provides a common means for identifying the status of components.
- **Notification Services**  
Provides a flexible way for components to notify interested parties of events, such as the starting of a MAS. It also provides the means for components to register their interest in specific events.
- **Single System Image**  
Supports the distribution of requests to multiple CMASs and MASs and the consolidation of results.

Kernel Linkage also controls the interface between code running under the MVS TCBs (selectors) and code running under the CICS TCB (methods).

This component has an identifier of KNL and its module names have the characters XL in the fifth and sixth positions (for example, EYU0XLNE).

## Trace Services

Trace Services provides other CICSplex SM components with the ability to write trace records to the CICS trace table and trace data sets. Trace Services is also responsible for writing any trace records created by a MAS to the trace table and data set of the managing CMAS. Tracing is a key part of CICSplex SM serviceability. Because a failure could occur at any time during CICSplex SM processing, Trace Services initializes as early as possible and terminates as late as possible in CICSplex SM processing.

This component has an identifier of TRC and its module names have the characters XZ in the fifth and sixth positions (for example, EYU0XZPT).

## Message Services

The Message Services component provides a common facility for building and issuing MVS console messages. The fixed text of messages and the variable text fragments used for insertion are defined in prototype tables. Calling methods then ask for messages by number and insert the appropriate variable text. Message Services is also responsible for creating the consolidated message log called EYULOG and for writing MAS-generated messages to the managing CMAS.

## common components

This component has an identifier of MSG and its module names have the characters XM in the fifth and sixth positions (for example, EYU0XMSM).

## Common Services

The Common Services component provides basic system services such as GETMAIN, FREEMAIN, POST, and WAIT processing. By routing all requests for these services through a single component, most CICSplex SM modules are isolated from the real environment in which they run. As a result, a relatively few methods (those that make up Common Services) need to be aware of the details of how these services are requested. One of the Common Services subcomponents provides timing services using the control task that runs as an MVS TCB. Another subcomponent provides locking services, both local (within a CMAS or MAS) and global (between a CMAS and its local MASs).

This component has an identifier of SRV and its module names have the characters XS in the fifth and sixth positions (for example, EYU0XSCG).

## Data Cache Manager

The Data Cache Manager component implements logical cache storage for use by CICSplex SM components. Each component can request a cache allocation and can allocate cache blocks within it. Several additional services are also provided by the Data Cache Manager:

- A quickcell service to improve the performance of getting and freeing frequently used blocks of a fixed size.
- A comprehensive set of list manipulation services for creating and maintaining ordered lists of data.
- Support for alternate indexing of cache lists.

This component has an identifier of CHE and its module names have the characters XC in the fifth and sixth positions (for example, EYU0XCLA).

## Queue Manager

The Queue Manager component implements queues of data within a cache that is shared between a CMAS and all its local MASs. Queues are often used to communicate between different CICSplex SM methods when the data to be passed is a set. Records within a queue can be accessed either sequentially or directly by relative record number.

This component has an identifier of QUE and its module names have the characters XQ in the fifth and sixth positions (for example, EYU0XQGQ).

## Data Repository

The Data Repository component provides methods for creating, accessing, updating, and deleting data in the CICSplex SM data repository, which is the VSAM data set where system configuration and definition data is stored. This component provides referential integrity support for the data repository and ensures proper rollback if an operation is only partially successful. Within this component are the following subcomponents:

- The Application Programming Interface provides access to CICS system management information and enables external programs to invoke CICSplex SM services.
- The Managed Object Services translate requests for data, for example, requests from real-time analysis, into the method calls required to obtain the data.

This component has an identifier of DAT and its module names have the characters XD in the fifth and sixth positions (for example, EYU0XDGR).

## Communications

Communications is one of the most complex components of CICSplex SM. It is made up of many subcomponents that provide all the services for implementing CMAS-to-CMAS and CMAS-to-MAS communication. In addition to the Communications component, CICSplex SM makes use of MVS program call routines in the ESSS. For communication between a CMAS and its local MASs, these program call routines provide cross-memory services for more efficient communication.

| Communication between one CMAS and another CMAS, can use either CICS  
 | intersystem communication (ISC) or interregion communication (IRC) services  
 | (usually referred to as multiregion operation, or MRO). Because routing of  
 | messages around the CMAS network does not require the user to define path or  
 | routing information, a subcomponent of Communications maintains a dynamic  
 | topology of the network and determines routes as required.

The Communications component implements a simple model for all other CICSplex SM components, that of remote method call. A method merely builds a MAL and invokes Communications via the Access Services subcomponent, specifying the destination and type of processing required. Communications then transports the MAL and causes it to be run in the target locations. All data required for the remote running of a method is automatically transported as well. Because all methods and their MALs are clearly defined, Communications knows what data must be sent to the target and what data must be returned to the caller. The data that is transported can be simple data in a MAL itself, data pointed to by a MAL, or CICSplex SM queues or cache lists.

This component has an identifier of COM and its module names have the character C in the fifth position (for example, EYU0CSLT).

## common components

---

## Chapter 3. Identifying a problem

Before you can determine the cause of a problem, you need to collect as much information as you can about your system and the symptoms you are experiencing. The following sections raise some basic questions that will help you identify the important information.

As you go through these questions, make a note of any changes to your environment and of any unusual occurrences, regardless of whether you think they are relevant. Even if the conditions you observe do not at first appear related to the problem, information about them could be useful later if you have to carry out systematic problem determination.

---

### Has CICSplex SM run successfully before?

If CICSplex SM has not run successfully before, it is possible that the system has not been installed or set up correctly. Refer to these other books in the CICSplex SM library for information on installation and setup requirements:

- *CICS Transaction Server for z/OS Program Directory* (or other installation instructions)
- *CICS Transaction Server for z/OS Installation Guide*:

In particular, you might want to try running the installation verification procedures, (IVPs), which are described in *CICS Transaction Server for z/OS Installation Guide*. These procedures are designed to verify the correct installation of CICSplex SM libraries and components.

---

### Have any changes been made since the last successful run?

If CICSplex SM has run successfully in the past, review any changes that have been made to your data processing environment since that time. Think about your operating systems, CICSplex SM itself, the CICS systems it manages, the hardware they run on, and any related operational procedures.

- If an APAR or PTF was applied to any of your operating systems, CICS, or CICSplex SM, check for error messages related to the installation. Also check for any unresolved ++HOLD ACTION items associated with the SMP/E maintenance. If the installation of maintenance was successful, check with your IBM Support Center for any known APAR or PTF error.
- If a hardware modification was made, it may have affected the systems on which CICSplex SM runs or the connectivity between systems in a CICSplex.
- If your initialization procedures changed, check for messages sent to the console during CICSplex SM or CICS initialization. It could be that changes to JCL, CICS system initialization parameters, or CICSplex SM system parameters are causing a problem.
- If the configuration of one or more CICSplexes has changed, check the EYULOG consolidated message log for messages describing incorrect or incompatible definitions. For example, if you are migrating additional CICS systems to management by CICSplex SM, ensure that the topology definitions for the new systems have been added to the CICSplex.

### Are there any messages that could explain the problem?

Check to see if there were any unusual messages issued during CICSplex SM initialization or immediately before the problem occurred. Also check for any messages related to a CICS system that is being managed by CICSplex SM.

If you find any messages that you don't understand, refer to the appropriate messages manual for an explanation and a recommended course of action.

---

### Does the problem occur at specific times?

If the problem seems to occur only at specific times of the day, consider what's happening in the system at that time:

- How many MASs are active? Where are they located and how are they communicating with the CMAS that is managing them? Have any MASs or CMASs recently become active and begun communicating with other address spaces?
  - Could the problem be related to system loading? Is the number of MASs (with associated resource activity) at its peak? If your CICSplex SM environment extends across more than one time zone, remember that the time of peak system usage may vary.
  - What type of monitoring, workload management, or analysis definitions are in effect? Keep in mind that the use of time periods can cause definitions to automatically become active or inactive at preselected times of the day.
- 

### Does the problem affect specific parts of the environment?

If the problem seems to affect only certain parts of the CICSplex SM environment, consider what is unique about those parts. If, for example, just one CMAS is experiencing a problem, review its configuration definitions:

- What system parameters were used in its startup job?
  - What other CMASs does it communicate with?
  - What CICSplexes does it participate in managing?
- 

### Common types of problem

Refer to Chapter 12, "Abends, stalls, and bottlenecks," on page 105 if:

- An abend has occurred.  
CICSplex SM-generated console, job log, or TSO terminal messages indicate that an abend occurred and provide an abend summary.
- A stall has occurred.  
The system is not responding to users logged on (to the MAS) or through the EUI (MAS or CMAS), or the system is using no, or an abnormally low number of, processor cycles.
- A bottleneck has occurred.  
The system or EUI response is abnormally slow, or the system is using an abnormally high number of processor cycles.

Refer to Chapter 13, "Investigating output and system management problems," on page 109 if:

- Data in an EUI display is not as expected (entries are missing or incorrect).
- CICSplex SM's system-management functions are not working as expected.

Monitor or analysis definitions are not active, real-time analysis events are not occurring, or are not being resolved, or a workload is being routed incorrectly, for example.

## common problems

---

## Chapter 4. Sources of information

This chapter describes some sources of information that you should find useful in problem determination.

---

### Your own documentation

This is the collection of information produced by your enterprise about what CICSplex SM should do and how it is supposed to do it. It could include:

- Flowcharts or other descriptions of system processing
- Record of configuration and topology definitions
- Record of resource monitoring, real-time analysis, and workload management activity
- Trace profiles for CMASs and MASs
- Performance statistics

---

### Change log

An up-to-date change log can identify changes made in the data processing environment that may have caused problems with your CICSplex SM system. For your change log to be useful in problem determination, it should include the following information:

- Changes in the system hardware
- Changes to corequisite programs (MVS and CICS)
- Changes to CICS resource definitions
- Maintenance applied to MVS
- Maintenance applied to CICS
- Maintenance applied to CICSplex SM
- Changes in operating procedures

---

### Manuals

In addition to this manual, you may need to refer to other manuals in the CICSplex SM library and the libraries for related products. For a list of CICSplex SM manuals that may be useful for problem determination, see “CICSplex SM books for CICS Transaction Server for z/OS” on page 146.

Make sure that the level of any manual you refer to matches the level of the system you are using. Problems often arise from using either obsolete information or information about a level of the product that is not yet installed.

---

### Online diagnostic aids

Assuming you can sign on to CICSplex SM or CICS, there are several online tools for collecting data about a problem:

- CICSplex SM views that provide diagnostic information about:
  - CMAS and MAS status
  - Resource monitoring activity
  - Real-time analysis activity
  - Workload management activity
- CICS commands that produce data similar to CICSplex SM data.

## online diagnostic aids

- The CICSplex SM online utility transaction (COLU), described in Chapter 9, “Using the online utility transaction (COLU),” on page 57.
- The CICSplex SM interactive debugging transactions (COD0 and CODB), described in Chapter 10, “Using the interactive debugging transactions (COD0 and CODB),” on page 61.

---

## Messages

Messages are often the first or only indication to a user that something is not working. CICSplex SM writes error and informational messages to a variety of destinations:

- The system console or system log
- The CMAS or MAS job log
- The EYULOG transient data queue
- The SYSOUT data set
- A CICS terminal
- The TSO READY prompt
- The ISPF end-user interface

Messages can be issued for many different reasons:

- An inappropriate user action
- Improper product installation or setup
- An error in CICSplex SM code

For more information on interpreting CICSplex SM messages, refer to the *CICSplex System Manager Messages and Codes*, which describes the standard CICSplex SM message formats, identifies the components that issue particular messages, and provides detailed information about each message.

---

## Symptom strings

Any CMAS or local MAS can produce symptom strings in a system or transaction dump. Symptom strings describe a program failure and the environment in which the failure occurred. All CICSplex SM symptom strings conform to the RETAIN<sup>®</sup> symptom string architecture. They are stored as SYMREC records in the SYS1.LOGREC data set.

A symptom string provides a number of keywords that can be directly keyed in and used to search the RETAIN database. If you have access to the IBM INFORMATION/ACCESS licensed program, 5665-266, you can search the RETAIN database yourself. If you report a problem to the IBM Support Center, you are likely to be asked to quote the symptom string.

Although symptom strings are designed as input for searching the RETAIN database, they can also give you information about what was happening at the time the error occurred. This information might point to an obvious cause for the problem, or a promising area in which to start your investigation.

---

## LOGREC records

LOGRECs are records containing information about an abnormal occurrence within CICSplex SM. The records are written to the SYS1.LOGREC data set and are available for analysis after a failure.

The LOGRECs produced by CICSplex SM all contain the same data. The data includes extensive information about the state of CICSplex SM components in the failing address space at the time the LOGREC is written, such as:

- Identification of the failing module
- Module calling sequence
- Recovery management information

---

## Traces

The CICSplex SM trace facilities provide a detailed record of every exception condition that occurs. They can also be used to trace various aspects of component processing.

In CMASs and MASs, CICSplex SM writes user trace records to the CICS trace data set, as follows:

- If any local MAS is in communication with a CMAS, trace data is shipped from the MAS to the CMAS, and a full, formatted trace record is produced.
- If any local MAS is not in communication with a CMAS (either because the Communications component is not yet active or because there is a problem with Communications itself):
  - A full, formatted trace record is produced if the MAS itself is running CICS/ESA 4.1 (or later).
  - An abbreviated trace record is produced if the MAS is not running CICS/ESA 4.1 (or later).

**traces**

---

## **Part 2. Tools for problem determination**

This part describes the tools available for problem determination. Each of these tools is covered in depth in the following chapters.



---

## Chapter 5. Using trace

All CICSplex SM address space (CMAS), managed application system (MAS), and coordinating address space (CAS) components provide trace data. This chapter begins by describing the CMAS and MAS trace facilities. The CAS trace is described on page 34.

---

### The CMAS and MAS trace facilities

The following sections describe the CICSplex SM trace facilities and the type of information they provide about CMASs and MASs.

#### Tracing in a CMAS

The CICS internal trace facilities must always be active in a CMAS.

When a CMAS is initialized, CICSplex SM ensures that the CICS trace facility is active and the trace table is large enough. The trace table settings required by the CMAS, along with the CICS SIT options that you need to use in order to establish these settings, are in Table 2:

*Table 2. Trace table setting required by the CMAS*

Trace variable	Required setting	CICS SIT option
Internal trace	On	INTTR=ON
Trace table size	2MB	TRTABSZ=2048
Master trace	Off	SYSTR=OFF
User trace	On	USERTR=ON

If the CICS trace facilities cannot be activated with these settings, CMAS initialization is canceled and you receive an error message.

Additionally, the CICS AUXTRACE facility should be active (for user records only) in a CMAS. If this facility is not active when a problem occurs, it may be necessary to recreate the problem with the facility turned on.

#### Tracing in a MAS

The CICS trace facilities do not have to be active in a MAS. Provided CICSplex SM communication facilities are available, MAS trace records are sent to a connected CMAS for recording; the only exceptions are trace records written for the CICSplex SM communication facility itself. If communication is not available, or if you are diagnosing a problem in the MAS, you may need to activate CICS tracing in the MAS.

**Notes:**

1. Although it is not required, it is strongly recommended that internal and AUXTRACE facilities be active (for user records only) in a MAS. CICSplex SM writes only exception records in a MAS, unless other trace records are specifically requested.
2. If any local MAS is in communication with a CMAS, trace data is shipped from the MAS to the CMAS, and a full, formatted user trace record is produced.

## CMAS and MAS trace facilities

3. If any local MAS is not in communication with a CMAS (either because the Communications component is not yet active or because there is a problem with Communications itself):
  - A full, formatted trace record is produced if the MAS itself is running CICS/ESA 4.1 (or later).
  - An abbreviated trace record is produced if the MAS is running CICS/MVS. The abbreviated trace record provides the following information:
    - Trace point ID
    - Module ID
    - Debugging text

## Types and levels of tracing

Each CMAS and MAS component can make use of three types and up to 32 levels of tracing.

### Standard trace (levels 1 and 2)

Standard trace points are designed to track the normal processing path of a component. There are two levels of standard tracing, level 1 and level 2. Trace points of this type are provided by every CMAS and MAS component. However, standard tracing is not normally active because it can cause additional overhead.

#### Usage Note

Level 1 and 2 trace points should be activated only for a specific CMAS or MAS component and only at the request of customer support personnel.

Level 1 trace points are used for:

- Module entry and exit
- Message parameter lists

Level 2 trace points provide information to supplement a level 1 trace and they require level 1 tracing to be active for the same component. Level 2 trace points are used for:

- Major data structures, including parameter list data addresses
- Other significant events

**Note:** Level 1 tracing must be active in order for level 2 traces to be collected. If level 2 tracing is requested for a component where level 1 is not active, no level 2 trace data is collected.

### Special trace (levels 3–32)

Special trace points can be used by a component for special-purpose traces that are unique to its situation. Each CMAS and MAS component has levels 3 through 32 available for special tracing. These trace levels provide detailed internal information about the component. For example, trace level 16, called a timing trace, is used by some components to record how long a request took to complete. This type of trace data can be used to evaluate the performance of a component under specific conditions.

#### Usage Note

Level 3–32 trace points should be activated only for a specific CMAS or MAS component and only at the request of customer support personnel.

### Exception trace

Exception tracing is always performed by each CMAS and MAS component when it detects an exceptional condition. The goal of this type of trace is *first failure data capture*, to capture data that might be relevant to the exception as soon as possible after it is detected. All CMAS and MAS errors result in an exception trace entry. Exception tracing cannot be disabled and all exception trace points are always active.

---

## Controlling the amount of tracing in a CMAS or MAS

During normal CMAS and MAS processing all the standard and special trace levels (levels 1–32) are usually disabled. Exception tracing is always active and cannot be disabled.

You can turn tracing on for a specific CMAS or MAS component in one of the following ways:

- Specify system parameters on a CMAS or MAS startup job, as described in Appendix B, “System parameters for problem determination,” on page 141.
- Use the ISPF end-user interface to activate one or more levels of tracing dynamically while CICSplex SM is running.
- Use the COD0 transaction TRACE flag command as described in “Method-level debugging with COD0” on page 62.

## Using the end-user interface to control tracing

You can use the CMAS or MAS view to control the tracing that occurs in an active CMAS or MAS.

For example, if you want to change the trace levels for the CMAS called EYUCMS1A, do the following:

1. Issue the CMAS view command.
2. Either issue the TRACE primary action command from the COMMAND field, as shown in Figure 6, or enter the TRA line action command in the line command field next to EYUCMS1A.

**Note:** To change the trace levels for a MAS, use the UPD line action command and then scroll to the Control MAS Trace input panel.

```

27FEB2005 14:46:30 ----- INFORMATION DISPLAY -----
COMMAND ==> TRACE EYUCMS1A SCROLL ==> PAGE
CURR WIN ==> 1 ALT WIN ==>
W1 =CMAS=====EYUCMS1A=EYUCMS1A=27FEB2005==14:46:30=CPSM=====2=====
CMD Name      Status  Sysid Access  Transit Transit
-----
Type----- CMAS---- Count--
EYUCMS1A ACTIVE  CM1A LOCAL          0
EYUCMS1B ACTIVE  CM1B ADJACENT      0
    
```

Figure 6. Controlling tracing from the CMAS view

3. The Component Trace input panel appears, as shown in Figure 7. It identifies the current trace settings for each component in the CMAS. A setting of Y means that trace level is active for the specified component; a setting of N means tracing is not active.

## controlling the amount of trace

```

----- Component Trace Levels for EYUCMS01 -----
COMMAND  ==>

      Overstrike the level number with a Y or N to alter the trace level

      Level
      1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3 3
      1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
Component -----
KNL      N N N N N N N N N N N N N N N N N N N N N N N N N N N N
TRC      N N N N N N N N N N N N N N N N N N N N N N N N N N N N
MSG      N N N N N N N N N N N N N N N N N N N N N N N N N N N N
SRV      N N N N N N N N N N N N N N N N N N N N N N N N N N N N
CHE      N N N N N N N N N N N N N N N N N N N N N N N N N N N N
QUE      N N N N N N N N N N N N N N N N N N N N N N N N N N N N
DAT      N N N N N N N N N N N N N N N N N N N N N N N N N N N N
COM      N N N N N N N N N N N N N N N N N N N N N N N N N N N N
TOP      N N N N N N N N N N N N N N N N N N N N N N N N N N N N
MON      N N N N N N N N N N N N N N N N N N N N N N N N N N N N
RTA      N N N N N N N N N N N N N N N N N N N N N N N N N N N N
WLM      N N N N N N N N N N N N N N N N N N N N N N N N N N N N
BAS      N N N N N N N N N N N N N N N N N N N N N N N N N N N N
Press Enter to change Trace Flags.
Type END to cancel without changing.

```

Figure 7. The Component Trace input panel

4. To change a trace setting for a specific component, such as Kernel Linkage (KNL):
  - a. Position the cursor next to KNL.
  - b. Move the cursor across the line to the appropriate level (1–32).
  - c. Type either Y, to activate tracing, or N, to deactivate tracing.
5. When all the trace settings are correct, press Enter. The CMAS view is redisplayed.

## Interpreting CMAS and MAS trace entries

A single CMAS or MAS trace can produce multiple records. Each record consists of a standard header followed by up to 3900 bytes of unique trace data. Within that data, each CMAS and MAS component uses a unique set of trace point IDs. Each trace point ID is used by only one trace point. A trace point ID consists of:

- Component ID
- Method ID
- Trace point number

Trace point numbers are assigned as follows:

<b>Range</b>	<b>Type of trace</b>
<b>0001–1024</b>	Exception trace
<b>1025–2048</b>	Level 1 trace
<b>2049–3072</b>	Level 2 trace
<b>3073–32767</b>	Special trace (Levels 3–32)

---

## Formatting CMAS and MAS trace entries

The CICSplex SM trace format utility, EYU9XZUT, formats the raw trace records produced for a CMAS or MAS. The host version of EYU9XZUT, formats the AUXTRACE records produced for a CMAS or MAS (including trace records sent to the CMAS).

### Trace formatting options on the host

The EYU9XZUT trace format utility has options that allow you to select the specific trace records to be formatted. You specify the formatting options you want to use on the SYSIN statement of the program's JCL, as described in "Trace formatting JCL" on page 33.

When no options are specified, all trace records in the trace data set are formatted.

EYU9XZUT supports the following options:

#### **ABBREV**

Provides an abbreviated trace, which has one line per trace record with a sequence number at the far right. Use the sequence number to select full trace formatting of specific records.

The abbreviated trace is written to a SYSOUT file named TRCEABB. You must provide a DD statement for this file when you request an ABBREV trace. If you do not provide the DD statement, an error message is produced and processing stops.

#### **COMPID=xxx,...|ALL**

Specify the three-character identifier of the components whose trace entries you want to format, or ALL for all CICSplex SM components. For a list of component identifiers, see "Major components of CICSplex SM" on page 139.

#### **EXCEPTION=ONLY|ALL**

ONLY formats only those exception trace records that match all other criteria. ALL formats all exception trace records, as well as any other trace records that match all other criteria.

**FULL** Provides full trace formatting of trace records meeting all selection criteria.

The trace is written to the SYSOUT file named TRCEOUT. You must provide a DD statement for this file when you request a FULL trace. If you do not provide the DD statement, an error message is produced and processing stops.

#### **METHOD=xxxx,...|ALL**

Specify the four-character identifier of specific methods whose trace entries you want to format, or ALL for all the methods for a component.

If the trace entries for one or more specific methods are required, customer support personnel will provide you with the appropriate method IDs.

#### **NAME=**

Specify the 1- to 8-character name of a CMAS or MAS whose trace entries you want to format.

The name appears on the trace heading, following the heading NAME.

#### **RECOVERY=ONLY|ALL**

ONLY formats only abend trace records, regardless of any other criteria that

## formatting trace entries

may be specified. ALL formats all abend trace records, as well as all trace records that match any other specified criteria.

**SEQ=** Specify one or more sequence numbers to select specific trace records.

The sequence number for each trace record appears at the far right of the formatted trace heading. Sequence numbers can be from 1 to 9 characters in length. A sequence number of zero is not valid.

Sequence numbers can be specified as a single entry or as a range of entries separated by a hyphen. For example:

SEQ=1-99,103,12345-12399

You can use up to 50 SYSIN cards with the SEQ= option. Each SYSIN data set can have up to 200 specific sequence entries, as either individual numbers or ranges. Any additional entries are ignored.

If you rerun the trace format utility using SEQ=, in order to get the same trace records you must specify all of the same options that you specified on the first run.

**TRANID=trn1,trn2,trn3....**

Specify the transaction ID of each transaction for which you want trace records.

The transaction ID appears in the formatted trace header, after TRANID.

**USER=**

Specify a TSO user ID.

The TSO user ID appears in the formatted trace header, after USER. Note that the USER= option is valid only for records that include an end-user interface unit of work.

You can request both an ABBREV and a FULL trace formatting in one run, by including both keywords in your SYSIN file and including the appropriate DD statements in the JCL.

### Hierarchy of formatting options

The combination of trace formatting options you select affects the output you receive. When you select:

#### **COMPID or METHOD**

Records for the specified component or method are printed.

#### **USER or NAME**

Records for the specified TSO user or system are printed.

#### **(COMPID or METHOD) and (USER or NAME)**

Only those records for the specified component or method that are also associated with the specified TSO user or system are printed.

#### **EXCEPTION**

When you specify ALL, all exception records are printed regardless of the other options you specify.

When you specify ONLY, exception records are printed for only the specified component, method, name, or user.

**SEQ** Selected records are printed, depending on the sequence of records you specify.

**TRANID**

Selected records are printed, depending on what you specify for all other options.

**Trace formatting JCL**

Figure 8 is an example of the JCL needed to run the host version of EYU9XZUT trace format utility.

```
//jobname JOB (acct),'name',CLASS=x,MSGCLASS=x
//TRCLST EXEC PGM=EYU9XZUT,REGION=2048K,PARM='NARROW'
//STEPLIB DD DSN=CICSTS31.CPSM.SEYULOAD,DISP=SHR
//SORTWK01 DD SPACE=(CYL,(3,2)),UNIT=SYSDA
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//TRCEIN DD DSN=cics.system.DFHTRACA,DISP=SHR
// DD DSN=cics.system.DFHTRACB,DISP=SHR
//TRCEOUT DD SYSOUT=*,COPIES=1
//TRCEABB DD SYSOUT=*,COPIES=1
//SYSIN DD *
ABBREVIATED
FULL
COMPID=MON
EXCEPTION=ONLY
METHOD=MSIN
RECOVERY=ALL
SEQ=1-55,77,999-1234567
TRANID=TRN1
/*
```

Figure 8. Example of JCL to execute the EYU9XZUT trace format utility

**Notes:**

1. The PARM='NARROW' parameter on the TRCLST EXEC statement causes the trace records to be printed in an 80-character format for display on a terminal. If you omit this parameter, the trace records are printed in their normal 132-character format.
2. The data set specified by the TRCEIN DD statement is the CICS auxiliary trace data set from a CMAS or a MAS.

```
                CVM.CICS.CVMSM2.DFHTRACB
CPSM Selective Trace Format Parameters:
ABBREVIATED=YES
FULL REPORT=YES
EXCEPTION=ALL
RECOVERY=ALL
TRANID=ALL
COMPID=WLM
METHOD=XCBA
NAME=ALL
USER=ALL
SEQ=ALL
EYU9XZUT - CICSplex/SM Trace Formatter
PROCESSING DATASET:CVM.CICS.CVMSM2.DFHTRACB
```

Figure 9. Example of output from the EYU9XZUT trace format utility (Part 1 of 3)

## formatting trace entries

```
TASK:00034 METHOD:XQLK PRIOR:XQGQ DEBUG:QLOKEXC POINTID: 2 TRANID:LP
MAJOB:QUE ENVRN:CMAS TYPE:EXCEPTION TOD:08:32:16.80730 CLO
UOW(CPSM): SYSTM:MVSH NAME:CICSCMH CICS-TASK: 34 TASK-STAR
```

```
MethName XQLK,XQGQ,CPTI,CPLT,XLOP
```

```
MAL          LEN:0030          ALET:00000000 ADDR:0502DBD0
EYUQXQLK IN
  *ENM FUNCTION( QUELOOK )
  CHR DEBUG( )
  *QID QTOKEN( 800C500600001920 )
  SDT DELETE( FALSE )
OUT
  *ENM RESPONSE( INVALID )
  *ENM REASON( QUEUE_ID_NOT_FOUND )
  *EPT CACHE_TOKEN( A= 00000000 O= 00000000 )
```

Figure 9. Example of output from the EYU9XZUT trace format utility (Part 2 of 3)

```
PROCESSING DATASETS: CVM.CICS.CVMSM2.DFHTRACA
```

```
TASK:00042 METHOD:XSWX ENVRN:CMAS TYPE:*ABEND* TRAN:MCCM TOD:05:24:31.48816
```

```
ABEND CODE: AEXY
PSW: 00000000 00000000
OFFSET: 6F6F6F6F
METHOD: XSWX
INTERUPT: 00000000 00000000
PROGRAM: EYU0XSWX
R04 06103D90 R05 00000008 R06 06104022 R07 06A37460
R08 06A36388 R09 05E64918 R10 05E65918 R11 06A3648C
R12 06A36080 R13 80045578 R14 85E6599E R15 85809080
ABEND ARREGS: AR00 00000000 AR01 00000000 AR02 00000000 AR03 00000000
AR04 00000000 AR05 00000000 AR06 00000000 AR07 00000000
AR08 00000000 AR09 00000000 AR10 00000000 AR11 00000000
AR12 00000000 AR13 00000000 AR14 00000000 AR15 00000000
ABEND STORAGE: -10 00000000 00000000 00000000 00000000
+00 00000000 00000000 00000000 00000000
```

Figure 9. Example of output from the EYU9XZUT trace format utility (Part 3 of 3)

---

## The CAS LU 6.2 communications trace

The following sections present information about the LU 6.2 communications trace and its interpretation.

### Tracing LU 6.2 communications

The BBC LU 6.2 component is responsible for LU 6.2 communications between address spaces (that is, between a CAS and another CAS, between a CAS and a CMAS, or between a CAS and a TSO user). The major portion of BBC resides only in the CAS; the remainder is loaded into any address space that is connected to the CAS.

The BBC LU 6.2 component keeps an internal wrap trace of all LU 6.2 messages sent to and from the CAS. This trace facility is always active and cannot be disabled. When there is a communications problem, such as a TSO user waiting for

a response, this trace can be helpful in determining the status of the response. Some of the information that can be found in the communications trace includes:

- User logons and logoffs
- Conversation allocations and deallocations
- Session connections and disconnections
- LU 6.2 communication protocol errors
- BBC messages issued to the operator console
- Requests for data

## Interpreting the LU 6.2 communications trace

The BBC LU 6.2 trace entries that you will find in a CAS dump formatted by the interactive problem control system (IPCS) can be viewed using the format described for the BBC LU6.2 Communication display.

The BBC LU 6.2 trace is made up of a X'10'-byte header followed by X'50'-byte table entries. The total length is X'7D10' bytes, which is usually sufficient to determine what, if anything, went wrong.

The format of the trace header is as follows:

**Offset Field**

- +0** MZZ\_ZRCF (Reserved field)
- +8** Current entry index
- +C** Last entry index

Figure 10 is an example of the first X'60' bytes of a BBC LU 6.2 trace:

```

BBCTRACE
LIST 7F6A52F0 ASID(X'0202') LENGTH(32016) AREA
ASID(X'0202') ADDRESS(7F6A52F0) KEY(48)
7F6A52F0. 00007D10 00000038 00000152 00000190 |..'.....|
7F6A5300. 008DC3C8 7F6A2C68 000005B6 00000000 |..CH".....|
7F6A5310. 00160000 00000000 00000000 00000000 |.....|
7F6A5320. 00000005 00000592 00000000 007212FF |.....k.....|
7F6A5330. 00380000 00000000 00000000 00000000 |.....|
7F6A5340. 0000D4E5 E2E35AD9 85869985 A288E389 |..MVST!RefreshTi|
    
```

Figure 10. BBC LU 6.2 trace record header

To locate the most recent entry, multiply the current entry index by X'50' (entry length) and add it to the first entry address of the table. Using the sample header shown in Figure 10, the following calculation determines the most recent entry in the trace:

$$\begin{aligned}
 X'152' * X'50' &= X'69A0' \quad (\text{Offset into the table}) \\
 X'7F6A5300' + X'69A0' &= X'7F6ABCA0' \quad (\text{Address of most recent entry})
 \end{aligned}$$

Each trace table entry has the following structure:

**Hex offset**

**Field**

- +00** TCB address
- +04** 0 for local message or address of outbound message
- +08** MZZ\_ZRCF (Reserved field)
- +10** Message type
- +14** Address of message in storage

## CAS LU 6.2 communications trace

- +18 Previous message
- +20 First X'30' bytes of message

**Note:** The message type (at offset +X'10') is an important piece of information for problem diagnosis. Customer support personnel may ask you to identify the message type in a trace entry.

---

## Web User Interface trace services

**Attention:** It is recommended that you activate trace only at the request of your IBM support center.

The CICSplex SM Web User Interface provides a trace service to help diagnose any problems encountered while using the interface.

You should ensure that you have specified the USERTR, SYSTR, and AUXTR CICS system initialization parameters in your Web User Interface server start-up job.

Trace records are written to the local AUXTRACE only and are not sent to the CMAS. These trace records can be formatted by the standard CICSplex SM trace formatter, EYU9XZUT.

You can control the amount of trace information produced by the Web User Interface by setting appropriate trace flags. Thirty one independent trace flags are provided and these can be enabled by:

- Using the WUITRACE Web User Interface system initialization parameter in the Web User Interface server start-up job
- From the COVC transaction

For more information about using trace and the EYU9XZUT utility, see the *CICSplex SM Problem Determination*.

## Setting trace flags using the WUITRACE parameter

You specify the WUITRACE parameter as described in *CICSplex System Manager Web User Interface Guide*. For example, to activate trace levels 13,15, and 31 when the Web User Interface is started, specify the WUITRACE parameter as follows:

```
WUITRACE(13,15,31)
```

## Setting trace flags through COVC

If you run the COVC transaction and select Trace Flags, you are presented with Figure 11 on page 37.

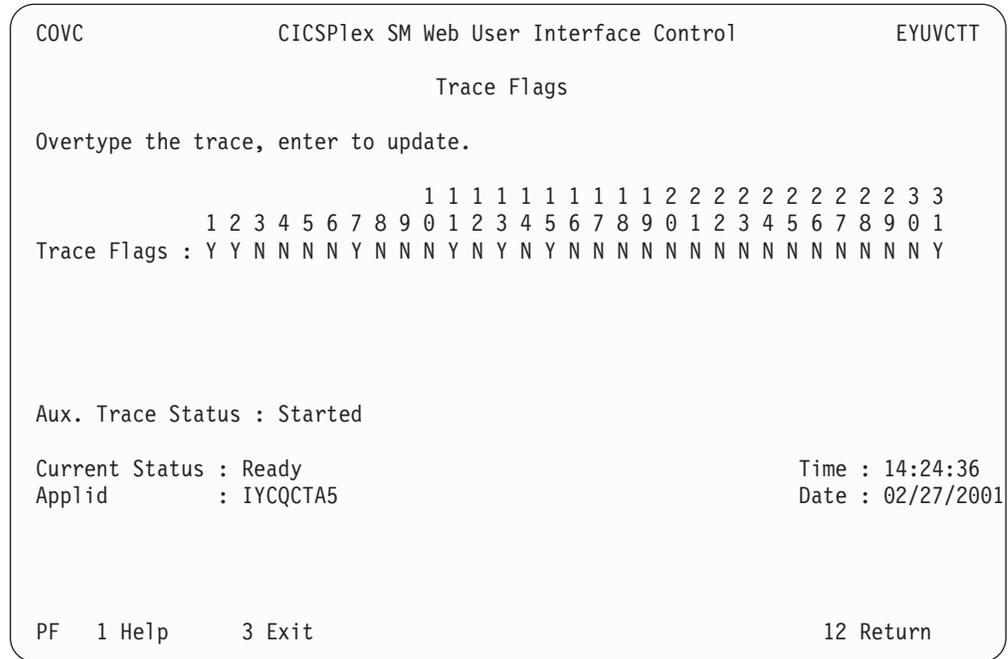


Figure 11. Trace flags

Figure 11 shows that trace levels 1,2, 7, 11, 13,15, and 31 are active. You can overtype any of the trace flags with Y or N to change the default settings.

**Note:** If you change any of the settings using the COVC transaction then restart the Web User Interface, the trace levels will be reset according to the levels specified on the WUITRACE parameter.

## The available trace flags

Table 3 shows a list of some of the available trace levels and their usage.

Table 3. Web User Interface trace levels

Level	Usage
1	Method entry and exit (summary)
2	Method entry and exit (detail)
3	Method entry and exit (special)
7	Stack management
8	Storage management
9	Service queue processing
10	Service event processing
11	View Editor
12	Resource catalogue
13	Stub management
14	View cache
15	HTTP requests and responses
18	API command
31	Soft exceptions

### Exception traces

Exception tracing is always performed by the Web User Interface server when it detects an exceptional condition. The goal of this type of trace is first failure data capture, to capture data that might be relevant to the exception as soon as possible after it has been detected.

Exception tracing cannot be disabled and all exception trace points are always active.

First-failure data capture is provided in two ways, as follows:

- Unexpected CICS and CICSplex SM responses and other detectable error conditions will result in non-maskable trace records being written.
- Errors such as program checks and abends will also result in a system dump and abend operator messages.

---

## Chapter 6. Using dumps

CICSplex SM can produce the following types of dump:

- Unexpected dumps
- CICSplex SM-requested dumps
- User-requested dumps

This chapter describes each of the three types of dump, and provides some guidelines for managing dumps.

---

### Unexpected dumps

Because CICSplex SM has a presence in two major parts of your environment, MVS and CICS, unexpected dumps may be produced at either level.

### CICSplex SM dumps under MVS

MVS can cause a dump to be taken for a CICSplex SM component in the CAS, a CMAS, or a TSO address space. This type of dump is an indication of a problem with the end-user interface.

When an unexpected dump occurs under MVS, CICSplex SM writes an abend indication and summary to either the console and job log, or to the TSO terminal. The first message in the abend summary is usually:

```
BBMZE001E Recovery manager ESTAE entered
```

CICSplex SM also produces LOGREC records related to the abend and, when appropriate, one of the following dumps:

- SDUMP
- SYSMDUMP, SYSUDUMP, or SYSABEND

Figure 12 is an example of a CICSplex SM dump produced under MVS.

```
BBMZE001E Recovery manager ESTAE entered
BBMZE002E ABEND S00C4-00000011 for ASID=0034-P; 0034/0034/0034,
BBMZE002E CSECT=MX Y10 (MX Y13+0A46)
BBMZE003E Error PSW=078D1000 89608962, IL/CC/IC=04/01/0011, LTA=7F4AD000
BBMZE004E Data at PSW-0A 47F0100A DD003000 F00044E0 10040520 47702008 4113E001
BBMZE005E Error registers 00-03 7F68B730 89608958 00000010 7F4AD136
BBMZE005E ..... 04-07 09609A60 09607F1C 7F68B7D0 7F68B7D0
BBMZE005E ..... 08-11 7F4C10D8 7F68B7D0 7F68B7D0 00000002
BBMZE005E ..... 12-15 7F6852E0 7F68B640 0000008A 007719A0
BBMZE005E RB registers 00-03 00000000 00000000 00000000 00000000
BBMZE005E ..... 04-07 00000000 00000000 00000000 00000000
BBMZE005E ..... 08-11 00000000 00000000 00000000 00000000
BBMZE005E ..... 12-15 00000000 00000000 00000000 00000000
BBMZE005E Retry registers 00-03 7F68B730 7F68BA50 00000010 7F4AD136
BBMZE005E ..... 04-07 09609A60 09607F1C 7F68B7D0 7F68B7D0
BBMZE005E ..... 08-11 7F4C10D8 7F68B7D0 7F68B7D0 00000002
BBMZE005E ..... 12-15 7F6852E0 7F68B640 0000008A 007719A0
BBMZE006E Linkage trace MXY13,MXBB2,MXB62,MXB71,MQ4S2,MXB65,MXBG4,MXCB2,MXC74
BBMZE006E ..... MXPMAIN,MXL01,MXUC2,MXG30,GS I=0100041E,MXL04,MXDRX
BBMZE006E ..... MXZR X
BBMZE802I Recovery manager ESTAE 'SDUMP' complete - RC=x'0000'
BBMZE801I Recovery manager ESTAE requested LOGREC recording
BBMZE803I Recovery manager ESTAE requested retry
```

Figure 12. Sample MVS abend indication and summary

## using dumps

Each MVS SDUMP has a title that consists of a summary of the abend. The title includes:

- The abend code
- The PSW at the time of the error
- The failing ASID
- The primary, secondary, and home ASIDs
- The failing CSECT
- The failing function and offset

Here is an example MVS/ESA SDUMP title:

```
BBM/ZE ABEND S0C4, PSW=078D1000 89608962, ASID=0034-P/0034/0034/0034,  
CSECT=MX Y10(MXY13+0A46)
```

In this example:

Abend code:	S0C4
PSW at time of error:	078D1000 89608962
Failing ASID:	0034
Primary, secondary, and home ASIDs:	0034, 0034, 0034
Failing CSECT:	MX Y10
Failing function and offset:	MX Y13 + X'A46'

## CICSplex SM dumps under CICS

CICS causes a dump to be taken for a CICSplex SM component when an abend occurs in a CMAS or MAS.

When an unexpected abend occurs under CICS, CICSplex SM writes an abend indication and summary to the console and job log. The first message in the abend summary is usually:

```
EYUXL0900I Starting environment recovery
```

CICSplex SM also writes a summary record to the CICS trace data set and takes a transaction dump, if appropriate. In addition, if the abend occurs in a CMAS or local MAS, CICSplex SM produces SYMREC records and, when appropriate, takes an SDUMP.

Figure 13 on page 41 is an example of a CICSplex SM dump produced under CICS.

```

+EYUXL0900I Starting Environment Recovery
+EYUXL0905E CICSCMH ASRB IN MCCD, OFFSET 000003D0 PSW=078D4000 8818A880
+EYUXL0905E INTC=0028 ILC=6 TXCP=0550D000 SCODE=S00E0 TRAN=MCCM TASK=0000041
+EYUXL0905E Methods=MCCD,MCCM,XLOP
+EYUXL0906I Registers at ABEND
EYUXL0907I GPR0-GPR3 05402EB8 05401178 00001FA8 0818A4F0
EYUXL0907I GPR4-GPR7 05401EB8 050271B0 0000000C 05400F10
EYUXL0907I GPR8-GPRB 003BE000 0547E6D8 053DC40C 0818B4F0
EYUXL0907I GPRC-GPRF 05400C88 05400F10 D8C3D900 07FD91E8
EYUXL0907I ARR0-ARR3 00000000 00000000 00000000 00000000
EYUXL0907I ARR4-ARR7 00000000 00000000 00000000 00000000
EYUXL0907I ARR8-ARRB 00000000 00000000 00000000 00000000
EYUXL0907I ARRC-ARRF 00000000 00000000 D4D6E2D5 00000000
+EYUXL0908I Storage At ABEND
EYUXL0909I -20 337E4199 00104660 336847F0 39C89AEE
EYUXL0909I -10 900058E0 9004B219 0200D203 D604E018
EYUXL0909I +00 5810D604 88100001 5010D608 B2190000
EYUXL0909I +10 D203D5FC 40105860 D55C9140 D5F047E0
+EYUXL0910I EYU9XLRV Dump,CICSCMH ,CICSCMH ,MVSH,CMAS,MCCM,0000041,
ASRB,EYU0MCCD,08/26/95,09:55:07
+EYUXL0999I XLRV Exiting Successfully

```

Figure 13. Sample CICS abend indication and summary

Each CICS SDUMP has a title that consists of a summary of the abend. The title includes:

- The name of the recovery routine that requested the SDUMP
- The MVS jobname
- The name of the CMAS or local MAS (as known to CICSplex SM)
- The 4-character MVS system ID
- The environment (CMAS or MAS)
- The CICS transaction ID
- The CICS task number
- The CICS abend code
- The full name of the CICSplex SM method that abended
- The date and time of the abend

Here is an example CICS SDUMP title:

```

EYU9XLRV Dump,CICSCMH ,CICSCMH ,MVSH,CMAS,MCCM,0000041,ASRB,
EYU0MCCD,08/26/98,09:55:07

```

In this example:

Name of the recovery routine that requested the SDUMP:	EYU9XLRV
MVS/ESA jobname:	CICSCMH
Name of the CMAS or local MAS:	CICSCMH
4-character MVS/ESA system ID:	MVSH
Environment (CMAS or MAS):	CMAS
CICS transaction ID:	MCCM
CICS task number:	0000041
CICS abend code:	ASRB
Full name of the CICSplex SM method that abended:	EYU0MCCD
Date and time of the abend:	08/26/98 09:55:07

## TRANDUMP and SYSDUMP code entries in a MAS

When the CICSplex SM local MAS agent starts, it automatically adds one CICS TRANDUMPCODE (TRANDUMP) entry for transaction dump code EYUN and two SYSDUMPCODE (SYSDUMP) entries for system dump codes EYU0XZPT and EYU0XZSD. These codes are primarily used for CICSplex SM Web User Interface

## using dumps

| users who many want to use the ADD action from the EYUSTARTTRANDUMP and  
| EYUSTARTSYSDUMP view sets to add their own TRANDUMP or SYSDUMP  
| entries.

| The presence of these entries does NOT indicate a problem, unless the current  
| dump count for these entries is greater than zero, in which case the MAS job logs  
| and dump may need to be investigated.

---

## CICSplex SM-requested dumps

A CICSplex SM-requested dump may be one of the following:

- An abend in the CAS
- A CMAS initialization failure
- A MAS initialization failure
- An abend in an Environment Services System Services (ESSS) program call (PC) routine

## Abends in the CAS

An SDUMP is scheduled for all abends in the CAS or any address space connected to it, except when:

- The DUMP=N parameter is specified on the CAS startup JCL
- The recovery manager receives control in nonprivileged state
- The abend is one of the following:
  - Sx06** Contents supervisor
  - Sx22** System CANCEL
  - Sx3E** DETACH
  - Sx7A** Contents supervisor
  - U4095** Task terminated by PGTERM

In addition to the failing address space, the master address space is always dumped so as to provide the MTRACE. If the abend occurred while in cross-memory mode, all the address spaces involved are also dumped. Because the CAS is often required to correctly diagnose a problem, it is always dumped, regardless of its involvement in the failure.

The CAS also provides DAE suppression support. Only one dump is taken for a specific set of symptoms. If it is a recordable failure, LOGREC entries are always written, but only one SDUMP is created. Symptom records are created and passed to SDUMP to prevent duplicate SDUMPs.

## CMAS initialization failures

If an abend occurs during CMAS initialization, the CMAS terminates. CICSplex SM takes an SDUMP with a dump code of EYUXL001 and writes a failure summary to the job log and console.

## MAS initialization failures

If an abend occurs during MAS initialization, the MAS agent code terminates; the CICS system continues to initialize, but it is not known to CICSplex SM. CICSplex SM takes a transaction dump with a dump code of EYUK and writes a failure summary to the job log and console.

## ESSS program call (PC) routine failures

If an abend occurs while a CICSplex SM PC routine is executing, the functional recovery routine (FRR) takes an SDUMP with a title whose format is as follows:

CICSplex SM (rrrr) Abend,(PC Set Name),(PC Routine Name),  
(Job Name),(SID),(date),(time)

where:

**rrrr** Is the release of CICSplex SM

### PC Set Name

Is the descriptive name of the set of PC routines that encountered the error.

It can be one of:

- Communication Services
- Dataspace Management
- Environment Services
- Lock Management
- MAS Assist Services

### PC Routine Name

Is the name of the PC routine within the set

### Job Name

Is the MVS Jobname

**SID** Is the MVS System ID

**date** Is the date in the form MM/YY/DD

**time** Is the time in the form HH:MM:SS

For each PC Set Name, the PC Routine Names are as follows:

### Communication Services

- ADDTHRD
- BINDAPI
- BINDEICB
- EADDTHRD
- EREMTHRD
- POSTECB
- REMTHRD
- SETAPI
- SETICT

### Dataspace Management

- CREATEDS
- DCMDS\_INFO
- DELETEDS
- EXCELETE
- EXDCMDS\_INFO
- EXDELGBL
- EXDELLCL
- EXEXTEND
- EXCREATE
- EXINFODS
- EXRELEASE
- EXTENDDS
- INFODS
- RELEASEDS

## using dumps

### Environment Services

APOTASK  
AUTHORIZE  
BIND  
CONNECT  
EAPITASK  
EXLCMAS  
EXLSIG  
EXRSIG  
FREE  
GSIGNAL  
IDENTIFY  
INQUIRE  
LISTCMAS  
LSIGNAL  
QUERY  
REGISTER  
RSIGNAL  
TERMINATE  
UPDPLEX

### Lock Management

ACQUIRE  
ADDLOCK  
EXACQLOCK  
EXADDLOCK  
EXRCVLOCK  
EXRELLOCK  
EXREMLOCK  
EXREMWAIT  
RCVLOCK  
RELEASE  
REMLOCK  
REMWAIT  
SSRCVLOCK  
SSREMLOCK

### MAS Assist Services

EMASINQ  
MASINQ

After it takes the SDUMP, the PC routine returns to its caller with a return code indicating that an abend occurred during processing.

---

## User-requested dumps

You can request a dump of a CICSplex SM address space at any time by:

- Using the MVS DUMP command
- Using the ISPF end-user interface

## Using the MVS DUMP command

You can issue the MVS DUMP command from the console to dump an Environment Services System Services (ESSS) address space, a CMAS, or a CICS/ESA or CICS/MVS MAS. Use the ASID= keyword to identify one or more address spaces and the DSPNAME= keyword to request data space dumps.

If you request a dump of data spaces, you must also dump the DMDS sysid data space and the ESSS address space, because that component owns all CICSplex SM data spaces. Data space names take the form:

cmpnsysid

where:

**cmp** Is either the three-character identifier of the component that uses the data space or DMDS, for the data cache master data space, which has controlling information for all data spaces. For a list of component identifiers, see “Major components of CICSplex SM” on page 139.

**n** Is the sequential number of a component data space.

**sysid** Is the four-character system ID of the associated CMAS.

**Note:** You can use the MVS DISPLAY ACTIVE command to display the ASID of the ESSS address space (EYUX140) and the names of data spaces.

A sample dump command for a CMAS might look like this:

```
DUMP COMM=(CMAS DUMP)
R xx,ASID=(2A,55),CONT
R xx,DSPNAME=(55.DMSHTC1,55.QUE1HTC1),END
```

## Using the ISPF end-user interface

You can use the SNAP action command on the CICSRRGN view to request a system dump for a MAS in those CICS releases that support it..

For example, if you want a system dump for the MAS called EYUMAS1A, do the following:

1. Issue the CICSRRGN view command.
2. Either issue the SNAP primary action command from the COMMAND field, as shown in Figure 14, or enter the SNA line action command in the line command field next to EYUMAS1A.

```
27FEB2005 11:30:30 ----- INFORMATION DISPLAY -----
COMMAND ==>> SNAP EYUMAS1A SCROLL ==>> PAGE
CURR WIN ==>> 1 ALT WIN ==>>
W1 =CICSRRGN=CICSRRGN==EYUPLX01==EYUCSG01==27FEB2005==11:30:30=CPSM=====3=====
CMD CICS Job MVS Act CICS CPU Page Page Total
--- System-- Name---- Loc Task-- Status Time---- In----- Out----- SIO-----
EYUMAS1A CICPRODA SYSA 34 ACTIVE 12345678 1234567 1234567 12345678
EYUMAS2A CICAOR1P SYSA 22 ACTIVE 567 1234567 1234567 10678
EYUMAS1B CICAOR2A SYSB 18 ACTIVE 10 1234567 1234567 25
```

Figure 14. Requesting a system dump from CICSRRGN

3. When the CICS SNAP input panel appears, as shown in Figure 15 on page 46, specify:
  - A 1- to 8-character dump code
  - An optional 1- to 8-character caller ID
  - An optional title of up to 79 characters

```
----- CICS SNAP -----  
COMMAND ===>  
  
Specify the options to be used for this dump of CICS:  
  
Dump Code ===> NORMAL          1- to 8-character dump code  
  
Caller      ===> NO            1- to 8-character caller ID  
  
                                TITLE (79 characters)  
  
Press Enter to continue CICS dump with the options specified.  
Type END or CANCEL to terminate dump request.
```

Figure 15. The CICS SNAP input panel

The following message appears in the window to confirm the dump request:

```
EYUEI0568I Dump Taken for EYUMAS1A, assigned DUMPID is nn/nnnn
```

where nn/nnnn is the dump ID assigned by MVS.

In the job log for EYUMAS1A you will see:

```
10.03.05 JOB00221 +DFHDU0201 EYUMAS1A ABOUT TO TAKE SDUMP. DUMPCODE: code, DUMPID: nn/nnnn  
10.03.12 JOB00221 IEA794I SVC DUMP HAS CAPTURED:  
                  DUMPID=005 REQUESTED BY JOB (EYUMAS1A)  
                  DUMP TITLE=CICSDUMP: SYSTEM=EYUMAS1A CODE=code ID=nn /nnnn  
  
10.03.05 JOB00221 +DFHDU0202 EYUMAS1A SDUMP COMPLETE.
```

---

## Managing dumps in MVS

Here are some general guidelines for managing system and transaction dumps:

- For MVS SDUMPs, either:
  - One or more SYS1.DUMPxx data sets must be defined.  
The size of these data sets depends on the amount of work being processed. Generally, SYS1.DUMPxx data sets in the range of 120–150 cylinders should be sufficient for use by CICSplex SM.
  - or
  - CICSplex SM can use dynamically allocated dump data sets when running under MVS/ESA 5.1 (or later).
- When an MVS SDUMP is not produced, SYSMDUMPs are the preferred alternative. Although this may not be practical for TSO sessions, it is strongly recommended that SYSMDUMP DD cards be included in all CMAS startup jobs. They should also be used for reproducible TSO session abends.

**Note:** No dump-related DD card is needed in the CAS startup job because the CAS takes only SDUMPs.

- When CICS transaction dumps are used to diagnose a problem, they must be formatted by the CICS utility DFHDUP directly from the original DFHDUMPx data set.
- The default size of the MVS internal trace table (64K) may not be large enough to contain the CICSplex SM trace data for some problems. It is recommended that the size of that table be increased to at least 256K.

## Using dumps with the Web User Interface

**Attention:** The Web User Interface server controller transaction COVC, should be used for debugging only at the request of your IBM support center. You must take steps to ensure that this transaction is used only by authorized personnel because of the access to system control areas that it provides. Improper or unauthorized use of COVC may have serious consequences, including without limitation loss of data or system outage. Customers are solely responsible for such misuse.

The CICSplex SM Web User Interface produces system dumps for all undetectable error conditions, including ASRA and AICA abends, but not transaction dumps.

### The available dump codes

The Web User Interface uses the dump codes described in Table 4:

*Table 4. Web User Interface dump codes*

Dump code	Description
EYU0VWAN	Unexpected abend in Web User Interface CICS Web Interface analyzer program (EYU9VWAN)
EYU0VWCV	Unexpected abend in Web User Interface CICS Web Interface converter program (EYU9VWCV)
WUIABEND	Unexpected abend in Web User Interface server program (EYU9VKEC)
WUITRACE	Dump requested by Web User Interface server trace point in program (EYU9VKEC)

## managing dumps

---

## Chapter 7. Displaying and formatting dumps with IPCS

The interactive problem control system (IPCS) provides MVS users with an interactive facility for diagnosing software failures. You can use IPCS to format and analyze SDUMPs produced by CICSplex SM or stand-alone dumps obtained while CICSplex SM was active in the system being dumped. You can either view the dumps at your terminal or print them. Workstation dumps cannot be handled by IPCS.

CICSplex SM provides a dump formatting routine that can be used with the VERBEXIT subcommand to format CMAS or MAS dumps.

For more information about:

- IPCS, see the *MVS Interactive Problem Control System: User's Guide*.
- Using IPCS to format CICSplex SM system dumps, see the *CICS Operations and Utilities Guide*.
- Preparing to use the IPCS tools, see *CICS Transaction Server for z/OS Installation Guide*.

---

### Using the CICSplex SM dump formatting routine

You can use the CICSplex SM dump formatting routine with the IPCS VERBEXIT command to analyze an SDUMP taken for a CMAS or MAS. The formatting routine lets you process a dump selectively by identifying one or more CMAS or MAS components as parameters to the exit.

The routine is supplied as module EYU9D210, but can also be identified to IPCS as CPSM210 when member CICSTS31.CPSM.SEYUPARM(EYUIPCSP) is installed. You can specify either name with the VERBEXIT command.

#### Usage Notes

- This dump formatting routine should be used only at the request of customer support personnel.
- If you are asked to send a copy of an SDUMP to support, you must send the unformatted dump.
- To distinguish between problems in the MAS agent code and problems in the underlying CICS system, support personnel may also ask you to format a MAS dump using the CICS DFHPDnnn formatting routine. For more information about this routine, see the *CICS Operations and Utilities Guide*.

### Formatting a CICSplex SM SDUMP

To format an SDUMP, use the following VERBEXIT command:

```
VERBEXIT CPSM210 'JOB=jobname,compid1,...,compidn,option,DLCT=nn...n,QID=nn...n'
```

where:

**JOB=** Identifies which CICSplex SM address space in the dump is to be formatted. If this parameter is omitted, the first CICSplex SM address space found is formatted.

## using the dump formatting routine

If no additional parameters are specified, the formatting routine does the following:

- Locates the selected address space.
- If the address space is a MAS, displays the CICS exit processing block.
- If DMDSxxxx data spaces are found in the dump, attempts to create equate symbol records (ESRs) for the data cache list tables (DCLTs) and all CICSplex SM ALET values.

The ESRs created by this routine can be used to display data space storage by the ALET value and offset found in any CICSplex SM control block. ESRs for the ALETs are in the form EYURECnnnnnnnn, where nnnnnnnn is the ALET value. To browse storage, use standard IPCS commands, such as L EYURECnnnnnnnn+yyyy, where yyyy is the offset into the data space.

### **compid1,...,compidn**

Identifies one or more specific CICSplex SM components for which dump data is to be formatted. If no component IDs are specified, only the CICSplex SM Kernel Linkage anchor block (XLWA) is formatted. For a list of component identifiers, see “Major components of CICSplex SM” on page 139.

For individual components, you can control the level of information that is produced by specifying `compid=n`, where `n` is one of the following:

- 1 Displays summary information, including a summary of CICS tasks for the component.

**Note:** For the Trace Services (TRC) component, this option formats only the exception trace records.

- 2 Displays detailed information, including the MODB, the MOEB, and all kernel linkage information for the component.
- 3 Displays both summary and detailed information for the component.

If no level is specified, both summary and detailed information are produced (or, in the case of Trace Services, all trace records are formatted).

**option** Requests additional noncomponent related information. The following options are supported:

**ESSS** Displays the ESSS address space control blocks.

### **LOCKS**

Displays a summary of resource locks.

**QLIST** Displays a summary of all allocated queues. Requires that the QUE dataspace(s) be present in the dump.

### **TASKS**

Displays a summary of CICS tasks for all CICSplex SM components.

### **DCLT=nn...n**

Identifies the DCLT to be displayed. DCLT identifiers, which are defined by the CPSM210 formatting routine as ESRs, can be from 12 to 16 bytes in length. (When the DCLT identifier is less than 16 bytes long, leading zeroes are assumed).

The DCLT control block and all elements associated with the DCLT are displayed. Each DCLT element is displayed as a separate block of storage.

**Note:** Both the data cache data space (DMDSxxxx) and the component data spaces containing cache list elements must be included in the

#  
#

dump for this routine to work properly. If the dump was produced by CICSplex SM as the result of an abend, the required data spaces may not be present. If, however, the data spaces are explicitly requested by a user, rather than by CICSplex SM, the processing should complete successfully. For more information on dumping data spaces, see “User-requested dumps” on page 44.

### **QID=nn...n**

Formats the selected data queue, showing the data queue service blocks, data queue record locate blocks, and the data queue record areas. The value nn...n is the 16-character data queue identifier.

The following is an example of a VERBEXIT command used to format dump data for specific components of a CMAS:

```
VERBEXIT CPSM210 'JOB=EYUCMS1A, TOP, RTA, MON=1, ESSS'
```

In this example, the address space to be formatted is EYUCMS1A. Dump data is produced for the Topology Services (TOP), real-time analysis (RTA), and Monitor Services (MON) components. For the Monitor Services component, only summary information is displayed. In addition to the component information, the ESSS control blocks are displayed.

### **CICSplex SM SDUMP summaries**

When the CPSM210 VERBEXIT is submitted to run in background, the following summaries are produced:

- Control block index, sorted by:
  - Area ID
  - Address space ID
  - Data space name
  - Location (either area address or data space offset)
- Control block index, sorted by
  - Address space ID
  - Data space name
  - Location (either area address or data space offset)
- Message index, containing the location of all messages.

The index contains a section of error message data and a section of informational message data. Each section contains a list of error messages sorted by message ID, and the page numbers of the output pages containing the message.

### **Formatting output for specific components**

You can specify the components for which you want to obtain SDUMP output.

- To obtain all available output for Monitor Services, real-time analysis, or Workload Manager, the format request must include the Topology Services component. Those components have areas anchored within the Topology Services control blocks.

For example:

```
VERBX CPSM210 'TOP, WLM'
```

- To obtain complete output, all data spaces associated with the selected components must be present in the dump.

For the Monitor Services component, the MAS1xxxx data space must be present, in addition to the TOP1xxxx and MON1xxxx data spaces; if it is not present, the output is incomplete.

**using the dump formatting routine**

---

## Chapter 8. Using the ESSS utility (EYU9XEUT)

The Environment Services System Services (ESSS) component of CICSplex SM is a limited function system address space that remains in the MVS image until the next IPL. ESSS implements a formal MVS subsystem for use by CICSplex SM.

This chapter describes how to use the batch utility program EYU9XEUT to perform diagnostic and maintenance functions on ESSS and the MVS subsystem.

### Usage Note

This utility program should be used only at the request of customer support personnel.

---

## The EYU9XEUT options

The EYU9XEUT batch utility program supports these options:

- DUMP
- RELOAD
- TERMINATE

You specify the option you want to use on the SYSIN statement of the program's JCL, as described in "The EYU9XEUT JCL" on page 54.

## Dumping data structures (DUMP)

The DUMP option reports on the contents of data structures in both the ESSS and the MVS subsystem at the time the program is run.

The format of the DUMP option is:

```
DUMP VERSION(nnn|ALL) [SUBSYSTEM] [ESSS] [LOCKS] [NOCML]
```

where:

### VERSION

Identifies the version of CICSplex SM for which a report is to be generated. *nnn* is a specific version of CICSplex SM, such as 210 for CICSplex SM for CICS Transaction Server for z/OS, Version 3 Release 1. ALL reports on each version of ESSS that has been created at your enterprise.

### SUBSYSTEM

Limits the report to the MVS subsystem data structures.

### ESSS

Limits the report to the ESSS data structures.

### LOCKS

Produces a summary of the ESSS data structures used by CICSplex SM locks.

### NOCML

Prevents EYU9XEUT from trying to obtain the MVS cross-memory local lock (CML), which may be held by a program call routine.

**Note:** No CICSplex SM lock summary is produced when NOCML is requested.

By default, the DUMP option generates a report containing MVS subsystem and ESSS data structures.

### Reloading broadcast functions (RELOAD)

CICSplex SM uses two MVS subsystem broadcast functions, end-of-task (EOT) and end-of-memory (EOM). As a result of program maintenance, it may be necessary to reload these functions in an existing ESSS address space.

The RELOAD option loads new broadcast functions from the utility library into the extended common system area (ECSA). You specify the location of the new functions on the UTILLIB statement of the program's JCL, as described in "The EYU9XEUT JCL."

The format of the RELOAD option is:

```
RELOAD VERSION(nnn) EOT|EOM|ALL
```

where:

*nnn*

Identifies the version of CICSplex SM for which broadcast functions are to be replaced. For example, specify 210 for CICSplex SM for CICS Transaction Server for z/OS, Version 3 Release 1.

**EOT|EOM|ALL**

Identifies the broadcast function to be replaced as end-of-task (EOT), end-of-memory (EOM), or both (ALL).

### Stopping the ESSS (TERMINATE)

The ESSS address space may occasionally need to be stopped to pick up changes made by CICSplex SM maintenance (PTFs), or when directed by IBM support personnel. The TERMINATE option requests that the ESSS address space is stopped. It may only be used when no other address spaces (for example, CMASes, MASes, CPSM API programs) are connected to the ESSS. To check that no address spaces are connected to the ESSS address space, use the EYU9XENF utility.

**Note:** Before using the TERMINATE option, all CICSplex SM CMASes, MASes, and CICSplex SM API programs using the same version of CICSplex SM as the ESSS MUST be stopped.

The format of the TERMINATE option is :

```
TERMINATE VERSION(nnn)
```

where:

*nnn*

Identifies the version of CICSplex SM for which the ESSS is to be stopped. For example specify 220 for CICSplex SM Version 2 Release 2.

---

### The EYU9XEUT JCL

Figure 16 on page 55 is an example of the JCL needed to run the EYU9XEUT utility program.

```

//jobname JOB (acct),'name',MSGCLASS=x
//UTIL EXEC PGM=EYU9XEUT
//STEPLIB DD DSN=data.set.name,DISP=SHR
//UTILLIB DD DSN=data.set.name,DISP=SHR
//SYSPRINT DD SYSOUT=*
//UTLPRINT DD SYSOUT=*
//SYSIN DD *
RELOAD VERSION(210) EOT
/*

```

Figure 16. Sample JCL for EYU9XEUT – RELOAD option

In this example, the RELOAD option is being used to load a new EOT broadcast function into the ECSA. The UTILLIB statement names the data set where the new broadcast function resides.

**Note:** To use this JCL for the DUMP option, delete the UTILLIB statement and change the RELOAD statement to a valid DUMP statement.

---

## Using the ESSS Information Display Utility (EYU9XENF)

The Environment Services System Services (ESSS) information display utility is a TSO/E command processor that can be used to display information about a CICSplex SM ESSS. It is typically used to ensure that no CICSplex SM address spaces (for example, CMASes, MASes, and CICSplex SM API programs) are connected to the ESSS. Before EYU9XENF can be used, it needs to be defined as a TSO/E authorized command, and available to the TSO user as follows:

- Add EYU9XENF to the 'AUTHCMD NAMES' section of the IKJTSoxx SYS1.PARMLIB member used by the MVS image.
- Issue from TSO a 'PARMLIB UPDATE(xx)' command to implement the changes made to the IKJTSoxx member.
- Ensure that CPSM310.SEYUAUTH is in the TSO users STEPLIB concatenation.

For details of IKJTSoxx and defining an authorized command see the *z/OS MVS Initialization and Tuning Reference*.

The format of the EYU9XENF command is :

```
EYU9XENF nnn
```

where:

*nnn*

Identifies the version of CICSplex SM for which the ESSS information is to be displayed. For example specify 310 for CICSplex SM Version 3 Release 1.

Figure 17 on page 56 shows an extract from a report produced by the EYU9XENF TSO command. From this figure we can see that two CMASes are active (in the ESSS), CMASPROD and CMASTEST. In addition three MASes are active PRODMAS1, PRODMAS2, and TESTMAS1.

**Note:** The CMAS information will only be removed after the CMAS has shutdown and all associated MASes and CICSplex SM API programs have stopped.

```

Version 310 CICSPlex SM Connection Information -----
  CMAS      Job/User  CICS      MAS      Job/User  CICSPlex
-----
CMASPROD   CMASPROD   PROD      PRODMAS1 PRODMAS1  PLEXPROD
           CMASPROD   PROD      PRODMAS2 PRODMAS2  PLEXPROD
CMATEST    CMATEST    TEST      TESTMAS1 TESTMAS1  PLEXTST
Version 310 CICSPlex SM ESSS Program Information -----
Program Name   Version   Load Point   Date       Time
EYU9X310      310      07C00AD0     12/21/00  19.59
EYUTXEPC      310      06755F90     12/21/00  19.49
EYU9XEER      310      00C1E158     09/21/00  10.55
EYU9XEEM      310      06A465D8     09/21/00  10.55
EYU9XEET      310      06A46170     07/01/00  12.32
EYU9XEEE      310      06753B10     07/21/00  19.59
Version 310 CICSPlex SM ESSS Resource Usage Information ---

```

Figure 17. Extract from ESSS information report from EYU9XENF

---

## Chapter 9. Using the online utility transaction (COLU)

The CICSplex SM online utility (COLU) is a CICS transaction that can be used to generate reports about various CMAS and local MAS components.

### Usage Note

This online utility should be used only at the request of customer support personnel.

---

### The COLU transaction

To run the CICSplex SM online utility, log onto a CICS system that is either a CMAS or a local MAS and enter the following transaction:

```
COLU compid keyword
```

where:

*compid*

Is one of the following 3-character component identifiers:

<b>CHE</b>	Data Cache Manager
<b>COM</b>	Communications
<b>KNL</b>	Kernel Linkage
<b>QUE</b>	Queue Manager
<b>SRV</b>	Common Services
<b>TOP</b>	Topology Services

*keyword*

Is a valid keyword for the specified component. Valid component and keyword combinations are described in the remainder of this chapter.

### Valid keywords for component CHE

#### CACHE

Summarizes the data space usage of each CMAS component. This keyword can be issued only from a CMAS.

#### LIST

Summarizes the data cache list usage of each CMAS component. This keyword can be issued only from a CMAS.

Figure 18 on page 58 is an example of the report produced by the CACHE keyword.

## COLU keywords

```
CICSplex SM 210 CICS/ESA SNAP Utility For JOB CVMCJBC 02/02/01
CPSM 210 DATA CACHE Dataspace Element Summary
CMAS Name: CMAS1JB Date/Time: 2/09/98 06:28:26.40770
Name ALET Start End Used
DMDSJWB1 01FF0013 00000000 00801000 00406E80
WLM1JWB1 01FF0017 00000000 00400000 000CA000
RTA1JWB1 01010042 00000000 00400000 00080000
MON1JWB1 01010041 00000000 00400000 00080000
TOP1JWB1 01FF0016 00000000 00400000 000A0000
COM1JWB1 01010040 00000000 00400000 000A0000
MAS1JWB1 01FF0015 00000000 00400000 000E4000
DAT1JWB1 0101003F 00000000 00400000 00175000
QUE1JWB1 01FF0014 00000000 00400000 00108000
```

Figure 18. Sample CACHE report from COLU

The CACHE report produced by COLU names the data space for each component of the CMAS and shows its ALET, its location, and the amount of storage used.

## Valid keywords for component COM

### MALRL

Lists all outstanding message argument lists (MALs) for the CMAS. This keyword can be issued only from a CMAS.

### MASRL

Lists all outstanding message argument lists (MALs) for all MASs attached to the CMAS. This keyword can be issued only from a CMAS.

### NETOP

Lists the communication network topology as it is known to the CMAS. This keyword can be issued only from a CMAS.

## Valid keywords for component KNL

### ESSSINFO

Summarizes the resources in use by the Environment Services System Services (ESSS) address space. This keyword can be issued only from a CMAS.

Figure 19 on page 59 is an example of the report produced by the ESSSINFO keyword.

```

CICSplex SM 210 CICS/ESA SNAP Utility For JOB CVMCJBC 02/02/01
      CPSM 210 Kernel Linkage CICSplex SM ESSS Connection Information
CMAS Name Job Name CICS SYSID MAS Name Job Name CICSplex Name
CMAS1JB CVMCJBC JWB1
CMAS1C3 CVMTC33 HTC3 CVMCT2 CVMCT2 PLEX1C1
CMAS1PP CVMCPPC PATC CSYS1PP CVMCPPM PLEX1PP
CMAS1JF CVMCJFC CMJF

CICSplex SM 210 CICS/ESA SNAP Utility For JOB CVMCJBC 02/02/01
      CPSM 210 Kernel Linkage CICSplex SM ESSS Program Information
Program Name Version Load Point Date Time
EYU9XESS 110 05A00E78 01/28/98 18.01
EYUTXEP 110 852A73E0 01/28/98 18.02
EYU9XEEM 110 855C1D38 02/02/98 03.06
EYU9XEET 110 8559C958 12/21/97 08.48

CICSplex SM 210 CICS/ESA SNAP Utility For JOB CVMCJBC 02/02/01
      CPSM 210 Kernel Linkage CICSplex SM ESSS Resource Usage Information
Resource Name Origin Length Number In Use
Connected ASID Table Elements 00006D98 0000D214 000001A4 00000006
Dynamic Work Area Elements 00013FAC 00010014 00000040 00000000
Lock Manager Resource Queues 0006BFD4 000A0014 00004000 0000031F
Lock Manager Holder/Waiter Elements 00023FC0 00048014 00002000 00000000
CICSplex Name Blocks 0010BFE8 00003014 00000400 00000004
Signal Blocks 0010EFFF 00006014 00000200 00000000

```

Figure 19. Sample ESSSINFO report from COLU

The ESSSINFO report produced by COLU provides information about active CMASs and the MASs that are connected to them, the ESSS system programs, and the ESSS resource tables.

## Valid keywords for component QUE

### ALL

Indicates that all allocated queues should be listed. When ALL is specified, no other keyword is permitted. This keyword can be issued from any CMAS or MAS.

### COMPID(*xxx*)

Is a 3-character CICSplex SM component ID. This keyword can be issued from any CMAS or MAS.

### METH(*xxxx*)

Is a 4-character CICSplex SM method name. This keyword can be issued from any CMAS or MAS.

### SUM

Causes a summarization report to be generated. In the detailed report, each line describes an allocated queue. This keyword can be issued from any CMAS or MAS.

Figure 20 on page 60 is an example of the report produced by the ALL keyword.

## COLU keywords

```
CICSplex SM 210 CICS/ESA SNAP Utility For JOB CVMCJBC 02/02/01
CPSM 210 Allocated Queue Resources
Queue Token AllcStg TotRec Meth MaxRecLn Mode Type DbugText
801B0001 00001060 00008000 00000000 XLNX 00000000 Del Wait NTFYQUE
801B0001 00001080 00008000 00000000 XDIN 00000000 Rept Work
801B0002 000010A0 00008000 00000004 CIIN 00000078 Rept Work EYU0CIIN
801B0002 000010C0 00008000 00000001 CIIN 00000052 Rept Work EYU0CIIN
801B0003 000010E0 00008000 00000001 CSSR 00000052 Rept Work COMMDEFS
801B0002 00001100 00008000 00000000 CWIN 00000000 Del Wait CWINXQCQ
801B0001 00001120 00008000 00000000 CPLT 00000000 Del Wait EYU0CPLT
```

Figure 20. Sample QUE ALL report from COLU

The QUE ALL report produced by COLU provides information about queue resources allocated by the CMAS or MAS, including their location, allocated storage, total number of records, method, maximum record length, mode, type, and text used in debugging.

## Valid keywords for component SRV

### LOCKS

Dumps the contents of all lock manager control blocks that are local to the CMAS or MAS. This keyword can be issued from any CMAS or MAS.

### LOCKSUM

Summarizes the lock manager usage of all locks that are local to the CMAS or MAS. This keyword can be issued from any CMAS or MAS.

Figure 21 is an example of the report produced by the LOCKSUM keyword. The LOCKSUM report produced by COLU provides information about local locks in

```
CICSplex SM 210 CICS/ESA SNAP Utility For JOB CVMCJBC 02/02/01
CPSM 210 Common Services Lock Management Summary
Resource Queue pool address 05F55014
Resource Queue pool size 00002800 ( 10K)
Total number of Resource Queues 256
Number of Resource Queues in use 5 ( 1.9%)
Resource Holder/Waiter Element pool address 05F57828
Resource Holder/Waiter Element pool size 00002400 ( 9K)
Total number of Resource Holder/Waiter Elements 256
Number of Resource Holder/Waiter Elements in use 0 ( 0.0%)

CICSplex SM 210 CICS/ESA SNAP Utility For JOB CVMCJBC 02/02/01
CPSM 210 Common Services Lock Management Summary
Lock 05F59F18 Owner CVMCJBC
Lock 05E18EA8 Owner CVMCJBC
Lock 00063228 Owner CVMCJBC
Lock 00062D28 Owner CVMCJBC
Lock 05DEA220 Owner CVMCJBC
```

Figure 21. Sample LOCKSUM report from COLU

use by the CMAS or MAS, including their location, size, and number.

## Valid keywords for component TOP

### PLEX(*plexname* [*,scope*])

Lists the topology of the specified CICSplex as it is known to the CMAS. The optional *scope* value limits the report to a named CICS system or CICS system group within the CICSplex. This keyword can be issued only from a CMAS.

---

## Chapter 10. Using the interactive debugging transactions (COD0 and CODB)

### ATTENTION

THE CICSplex SM INTERACTIVE DEBUGGING TRANSACTIONS COD0 AND CODB SHOULD BE USED ONLY AT THE REQUEST OF IBM CUSTOMER SUPPORT PERSONNEL. YOU MUST TAKE STEPS TO ENSURE THAT THESE TRANSACTIONS MAY BE USED ONLY BY AUTHORIZED PERSONNEL BECAUSE OF THE EXTENT OF THE ACCESS TO SYSTEM CONTROL AREAS THAT THEY PROVIDE. IMPROPER OR UNAUTHORIZED USE OF COD0 AND CODB MAY HAVE VERY SERIOUS CONSEQUENCES, INCLUDING WITHOUT LIMITATION LOSS OF DATA OR SYSTEM OUTAGE. CUSTOMER SHALL BE SOLELY RESPONSIBLE FOR SUCH MISUSE.

The interactive debugging transactions COD0 and CODB provide access to the CICSplex SM run-time environment. They can be used to format and manipulate the internal data structures of CICSplex SM. The debugging transactions can run in CMASs, and in CICS/ESA, and CICS/MVS, MASs with terminal support.

---

### Running the debugging transactions

To run the CICSplex SM debugging transactions, log on to a CICS system and enter one of the following transaction IDs:

**COD0** To use the method-level debugging transaction, as described in “Method-level debugging with COD0” on page 62. This transaction provides access to CICSplex SM objects, methods, message argument lists (MALs), and outstanding requests. To exit this transaction, type EXIT on the command line.

**CODB** To use the system-level debugging transaction, as described in “System-level debugging with CODB” on page 90. This transaction provides access to address space and data space storage, major control blocks, data queues, and CICSplex SM entries in the CICS trace table. To exit this transaction, press PF3 or type END on the command line.

The following usage rules apply to the COD0 and CODB transactions:

- You issue a COD0 command by typing the command name on the command line. You issue a CODB command by typing its option number on the command line.
- The standard END and CANCEL commands are recognized. END completes the task in progress and returns you to the previous screen, while CANCEL cancels the task before returning.
- You can scroll a display by using the commands DOWN, UP, TOP, and BOT. With COD0, you can also enter a default scrolling amount in the Scroll==> field.
- On a selection list, any character that is not a blank or an underscore can be used to select an option.
- These transactions support only 3270 model 2 screens that is, 24x80 and 32x80 type screens.

## Method-level debugging with COD0

After logging onto CICS, enter the COD0 transaction ID to display the COD0 main menu. Figure 22 shows the COD0 main menu displayed under CICS/ESA, and CICS/MVS.

To issue a COD0 debugging command, enter it in the CMD=> input field.

```

COD0 CICSplex/SM Debugger
CMD=>                               Scroll=> PAGE
Welcome to CICSplex/SM Debugger. Commands available are:

      ALLOC      Allocate storage, cache, cache list, queue, or eptr.
      ATTACH     Starts a method running in MAS/CMAS.
      CALL       Call a CICS transaction or program.
      CAPTURE    Capture an EUI View to print.
      DUMP       Call COB transaction to display memory.
      EXEC       Executes a method immediately from the debugger.
      EXIT       Exit the debugger.
      LIST       List methods, CP/SM tasks, and resources.
      POST       Posts an ECB using MVS POST.
      PRINT      Print a CP/SM data area to the JES Spooler.
      PURGE      Delete a resource ALLOCated.
      START      Starts a method running in CMAS.
      TRACE      Set CICS component trace flags.
      TRACK      Set CPSM trace flags based on calling structure.
      TRAP       Set tracing flags for a single method.

Enter HELP (command) for more help on commands.
    
```

Figure 22. COD0 debugging transaction menu (CICS/ESA, and CICS/MVS)

Commands can include one or more parameters, which must be separated by one or more spaces. Commas and quoted strings are not supported.

As in ISPF, PF key functions are prefixed to whatever is on the command line. The following PF keys are in effect when COD0 is running in a CICS/ESA, or CICS/MVS, environment:

Key	Description
PF1	HELP
PF3	END
PF4	PREV
PF5	NEXT
PF7	UP
PF8	DOWN

## Issuing commands recursively

You can enter the debugger commands recursively from any screen in the COD0 transaction, effectively nesting the commands and their output. However, under CICS/ESA, and CICS/MVS, when the LIST and HELP commands are entered recursively, the new output replaces the old. For example, if you issue the LIST START command followed by the LIST TASK command, the LIST TASK output replaces the LIST START output.

## Issuing commands that alter CICSplex SM

The following COD0 debugging commands can be used to modify memory or some other aspect of CICSplex SM operation:

- ATTACH
- EXEC

- POST
- START

When you issue one of these commands, you receive a warning and confirmation panel. You should proceed with the command only at the request of customer support personnel.

## ALLOC (allocating a resource)

The ALLOC command allocates a resource so that you can refer to it by name in completing MALs. The resource can be a cache list, a data queue, data space storage, or shared CICS storage.

The format of the ALLOC command is:

```
ALLOC /resname [optional parameters...]
```

where:

*/resname*

Identifies the resource being allocated. The resource name can be no more than eight characters, including the required slash.

The optional parameters are:

**QUEUE** *compid*

Creates a queue token and assigns it to the resource being allocated. *compid* is the 3-character component identifier, as listed in “Major components of CICSplex SM” on page 139.

**CLIST**

Displays the Allocate CACHE LIST input panel (shown in Figure 23 on page 64), which lets you create a CACHE LIST token and assign it to the resource being allocated.

**STG** *size* [**BELOW**]

Acquires an address of the specified size from CICS shared storage and assigns it to the resource being allocated. *size* is a number of bytes. The **BELOW** option requests storage from below the 16MB line; by default, storage is acquired above the line (in 31-bit mode).

**EPTR** *size*

Acquires a data space pointer of the specified size from a data space and assigns its **ALET** and **OFFSET** to the resource being allocated. *size* is a number of bytes.

```
COD0 CICSplex/SM Debugger
CMD=>                                                                    Scroll=> PAGE

Allocate CACHE LIST

  Id of CACHE to create CACHE LIST: /@CACHE          (Optional)

  Estimated number of elements:

  Element size:

  Estimated free space:          (Optional)

  GENERIC if generic keys:      (Optional)

  Hash Table Size:              (Optional)

  Key Offset: 0                  (Default 0)

  Key Size:

  Search method (BINSRCH/HASH): BINSRCH
```

Figure 23. Allocate CACHE LIST panel

Resources remain allocated across multiple COD0 transactions or between multiple COD0 transactions running concurrently in the same CICS system. In fact, all resources exist until you specifically purge them.

### Notes:

1. You can use the LIST ALLOC command to display a list of allocated resources.
2. You can use the DUMP /resname command to dump the storage, data queue, or cache list for an allocated resource.

## ATTACH (attaching a method)

| The ATTACH command starts a method running in the CICS systems identified by  
| the specified context and scope values.

The format of the ATTACH command is:

```
ATTACH method context scope
```

where:

*method*

Is the ID of a CICSplex SM method that has been coded to run from a CICSplex SM end-user interface request.

*context*

Is the name of a CMAS or CICSplex.

*scope*

Is the name of a CICSplex, CICS system group, or CICS system.

For a list of valid responses to this command, see “Running a method” on page 89.

| Unlike the START command, which merely starts a CICS transaction within a  
| CMAS, ATTACH crosses the boundary between a CMAS and a local MAS. (These  
| methods may run in the CMAS, a different address space, or even a different

processor in the CICSplex). To accomplish this, ATTACH uses the logic of an end-user interface request, which runs under its own TCB in the CMAS.

Figure 24 is an example of the display for a completed attached task that ran within a single CMAS or MAS.

Figure 25 is an example of the display for a completed attached task that either ran

```

COD0 CICSplex/SM Debugger
CMD=>                               Scroll=> PAGE
Enter END to exit or ENTER to view results.
Status for ATTACHed method XQCQ

    Methods status: Method completed.
    XLCI return description: OK
    Method's RESPONSE was:   OK
    Method's REASON was:

CONTEXT: CVMCTS01  SCOPE: CSYSGRP1  REGION: CSYSGRP1

Unit of work
  SYSID: TEST      USERID: DEVOPER TCB 00452160
  Major Object: 00  Component Id: 73

The method executed in a single MAS so all information
appears in the fields of the MAL.
    
```

Figure 24. Attached task display for a single CMAS or MAS

in multiple MASs, or ran multiple times in a CMAS.

When you press Enter, each of the MALs that ran in each region is reconstructed

```

COD0 CICSplex/SM Debugger
CMD=>                               Scroll=> PAGE
Enter END to exit or ENTER to view results.
Status for ATTACHed method XQCQ

    Methods status: Method completed.
    XLCI return description: OK
    Method's RESPONSE was:   OK
    Method's REASON was:

CONTEXT: CVMCTS01  SCOPE: CSYSGRP1  REGION: CSYSGRP1

Unit of work
  SYSID: TEST      USERID: DEVOPER TCB 00452160
  Major Object: 00  Component Id: 74

The method executed in multiple MAS so a queue of OUT records
was created.

OUTQUE QUEUE ID: A4957FBD B3E11932
                Records : 0000013 Record Length: 0000018
    
```

Figure 25. Attached task display for multiple CMASs or MASs

and displayed individually, as shown in Figure 26 on page 66.

```
COD0 CICSplex/SM Debugger
CMD=>
NEXT/PREV to browse CICS region MALS. END=Exit.
MAL for CICS Region:CICSSY01
IN
*ENM FUNCTION( CREQUE )
  CHR DEBUG(          )
  PTR ECB( 00000000 )
*CMP MAJOR_OBJECT( KNL )
*ENM TYPE( WORK )
*SDT DELETE( TRUE )
OUT
*ENM RESPONSE( OK )
*ENM REASON(          )
*ETK QTOKEN( A4957FC53998FB31 )
```

Scroll=> PAGE

Figure 26. MAL display for a specific CICS region

Note also that the region the MAL ran in is shown on the header line for the display. You can use the NEXT (PF5) and PREV (PF4) keys to browse backwards and forwards between the regions. END (PF3) returns you to the attached task display.

## CALL (calling external CICS programs and transactions)

The CALL command calls a CICS transaction or program with optional parameters.

The format of the CALL command is one of the following:

```
CALL cicstran [optional parameters...]
```

```
CALL PROGRAM cicsprog [optional parameters...]
```

**Note:** CICS can be used as a synonym for CALL.

*cicsprog*

Is a program ID that must be defined to CICS.

*cicstran*

Is a transaction ID that must be defined to CICS.

The parameters are passed as a TIOA area, so anything that can be entered at the transaction's or program's initial screen can be specified as an optional parameter. There is no validation of the optional parameters.

For transaction calls, the transaction ID is placed as the first field in the constructed TIOA (as it would be from the terminal). Make sure the transaction is defined as conversational. Pseudo- or nonconversational programs return immediately to COD0.

For program calls, you must enter the transaction ID as the first parameter, if the program you are calling expects this.

While the task is running, all the facilities of that CICS transaction or program are available to you. When you end the task, you return to COD0.

**Note:** You should not attempt to call:

- The COLU transaction, which is used by CICSplex SM
- CICSplex SM programs, which begin with the letters 'EYU'

## CAPTURE (capturing and printing a view)

The CAPTURE command captures and prints all communications related to a CICSplex SM end-user interface view being issued by a particular user. CAPTURE uses the CICS spool facility to write the data as an output file called Sxxxxxxx, where xxxxxx is a numeric identifier.

The format of the CAPTURE command is one of the following:

```
CAPTURE viewname userid count
```

```
CAPTURE *MASMON montype count
```

where:

*viewname*

Is the name of the CICSplex SM view to be captured.

*userid*

Is the TSO user ID of the user who will be issuing the view command.

*count*

Is the number of times the view should be captured.

A count is taken from the time the view command is entered until the user enters another view command or END. Pressing Enter repeatedly to refresh the data or perform some action against the view does not change the count of the view command.

You can reissue the CAPTURE command with the same view name and user ID to update the count. A count of zero deletes the CAPTURE entry.

**\*MASMON**

Captures monitor data as it is collected by a MAS.

*montype*

Is the type of monitor data to be captured:

**MCICS**

CICS regions

**MCONN**

Connections

**MDBX** DB2<sup>®</sup> and DBCTL resources

**MFILE** Files

**MGLBL**

Global resources

**MJRNL**

Journals

**MPROG**

Programs

**MTDQS**

Transient data queues

**MTERM**

Terminals

**MTRAN**

Transactions

For example:

```
CAPTURE MONDEF USER39 3
```

captures the next three MONDEF view commands issued by USER39. All related MALs and queues are printed.

## DUMP (displaying and altering data)

The DUMP command displays a scrollable dump of memory. Some parameters of the DUMP command cause the CICSplex SM system-level debugging transaction, CODB, to be invoked.

If you alter the displayed memory, you must enter UPDATE (or press PF11) to record the change. If you alter memory but do not enter UPDATE, a message is displayed to remind you to enter UPDATE.

The format of the DUMP command is:

DUMP [parameters...]

where the parameters are:

*/resname*

Displays the queue, EPTR, storage, or cache list allocated to the specified resource.

*@method*

Calls CODB with the entry point of the specified CICSplex SM method.

*hexadecimaladdress*

Assumes the hexadecimal value is an address and enters CODB with ALET=0 and the address specified.

*hexadecimalALET hexadecimaloffset*

Displays the address of the specified ALET (first hexadecimal value) at the specified offset (second hexadecimal value).

*hexadecimaladdress [length]*

Displays the storage starting at the specified address. The amount of storage displayed is determined by the length parameter. Length is assumed to be a decimal value, unless a X'length' value is specified.

**CACHE** *cachetoken*

Displays the data identified by the specified cache-list token. The token is entered as two 8-byte hexadecimal character strings. To display the previous record, use PF4; to display the next record, use PF5; to display a specific record, enter REC *n*, where *n* is the record number.

**CLIST** *token*

Calls CODB with the specified token. The token is entered as two 8-byte hexadecimal character strings.

**EIB** *compid*

Displays the address of the CICS information block for the first transaction running under the specified component.

**EIS** *compid*

Displays the address of the CICS storage block for the first transaction running under the specified component.

**MODB** *compid*

Displays the address of the MODB for the specified component.

**MODD**

Displays the address of the MODD.

**MOEB** *compid*

Displays the address of the MOEB for the specified component.

**OPB** *compid*

Displays the address of the first OPB for the specified component.

**QUE** *token*

Displays the data identified by the specified token. The token is entered as two 8-byte hexadecimal character strings.

**STAKEND** *compid*

Displays the address of the last stack for the first transaction running under the specified component.

**STAKSTR** *compid*

Displays the address of the initial stack for the first transaction running under the specified component.

**XLWA**

Displays the CICSplex SM kernel linkage work area.

Figure 27 shows an example XLWA display.

COMMAND==>	XLWA	COMP ID==>	ADDR==>	ALET==>	00000000
MSG==>					
0012D0A8	00000000	08006EC5	E8E4D5E7 D2D5D3C3	E6C1C1C2	..>EYUNXKNLCWAAB
0012D0B8	00000010	0D000200	00000000 00000000	00000000	.....
0012D0C8	00000020	8001608B	0000000E 0000EE00	11F20000	..-.....2..
0012D0D8	00000030	0012D168	0000DE5C 113E8000	00000546	..J....*.....
0012D0E8	00000040	116D4468	00040000 11F2EE00	00100000	.....2.....
0012D0F8	00000050	11F2EE00	FFFFFF6A 00000000	00000000	.2..... .....
0012D108	00000060	00000000	D2D3D7C2 116D8EB0	116DB328	....KLPB._..._.
0012D118	00000070	116D7728	00CA812C 00000000	00000000	_. ....a.....
0012D128	00000080	00000000	00000000 00000000	00346EC5	.....>E
0012D138	00000090	E8E4E7C5	C5E8E4D9 E7C5D3E2	01030000	YUXEEYURXELS....
0012D148	000000A0	00CC4008	0E1B9CE8 00CA8118	00CA812C	.. ....Y...a...a.
0012D158	000000B0	00F69A00	009C9520 0000011C	00000001	.6....n.....
0012D168	000000C0	00000000	11F20000 11F21100	11F22200	....2...2...2..
0012D178	000000D0	11F23300	11F24400 11F25500	11F26600	.2...2...2...2..
0012D188	000000E0	11F27700	11F28800 11F2AA00	11F29900	.2...2h..2...2r.
0012D198	000000F0	11F2BB00	11F2CC00 00000000	00000000	.2...2.....
0012D1A8	00000100	11F2DD00	00000000 00000000	00000000	.2.....
0012D1B8	00000110	00000000	00000000 00000000	00000000	.....
0012D1C8	00000120	00000000	00000000 00000000	00000000	.....

Figure 27. An example DUMP XLWA display

**Notes for CICS/ESA, and CICS/MVS, users:**

1. If you issue the DUMP command without parameters, the CODB main menu is displayed.
2. For CICSplex SM components, CODB displays the first transaction running under that component, which is its first OPB. You can use the NEXT and PREV PF keys to display multiple transactions. You can also use the LIST TASK command to display all of the stacks and methods in all of the CICSplex SM tasks, and then select specific stacks, methods, or OPBs to display.

**EXEC (executing a method)**

The EXEC command executes a method directly from the COD0 debugging transaction.

The format of the EXEC command is:

EXEC method

## COD0 transaction

where:

*method*

Is the name of a CICSplex SM method.

The formatted message argument list (MAL) for the method is displayed. For details about how to enter data from this display, see “Displaying a MAL from COD0” on page 86. For a list of valid responses to this command, see “Running a method” on page 89.

## EXIT (exiting COD0)

The EXIT command exits the COD0 debugging transaction. This command has no parameters.

You can use this command to exit the debugging transaction from any screen. A closing message is displayed; you can then clear the CICS screen and enter another transaction.

### Note for CICS/ESA, and CICS/MVS users:

All allocated resources and started or attached tasks are recorded in a temporary storage record. The next time you enter COD0, all allocated resources are still available and all started or attached tasks can be displayed using the LIST START command.

## HELP (getting online help)

The HELP command displays help text for COD0 commands.

The format of the HELP command is:

HELP [cmdname | COMPID]

where:

*cmdname*

Is the COD0 command for which help information is to be displayed.

**COMPID**

Produces a list of identifiers of CICSplex SM components.

If you issue the HELP command without parameters, the initial help panel, which lists all COD0 commands, is displayed.

## LIST (listing tasks and allocated resources)

The LIST command lists running CICSplex SM tasks, the status of started and attached tasks, and the allocated resources available to you.

The format of the LIST command is:

LIST [parameters...]

where the parameters are:

**ALLOC**

Lists all allocated resources. You can purge or dump resources from this screen.

**ATCB**

Lists the API task control blocks used for processing CICSplex SM API requests.

**CACHE**

Lists the data caches in use by a local MAS.

**CAPTURE**

Lists all ISPF end-user interface views currently being captured.

**CLIST**

Lists the data cache lists in use by a CMAS.

**COMM**

Lists the two communication MAL queues: one for methods executing via the CMAS, and the other for methods routed to a MAS.

**DLL [TASK *cicstaskid* |PID *processid*]**

Lists dynamic link libraries (DLLs) for the all CICSplex SM components loaded for a process. By default, the details are for the DLLs loaded for the process running the CONL transaction. However, on some operating systems these are not guaranteed to be the same for all processes. To display details for another process, specify either the CICS task ID or the operating system process ID.

**METH [*compid*]**

Lists all methods within the specified component. If no component is specified, all methods are listed.

**START**

Lists all started and attached tasks and their current status. You can purge, display, or dump the MAL created from this screen as well as restart, attach, or execute the same MAL.

**STCB [ERRORS]**

Lists the server-client control blocks. The ERRORS option provides a description of any errors encountered.

**TASK [*compid*]**

Lists the CICSplex SM tasks from the specified component showing all active method calls. If no component is specified, all tasks are listed.

**IEWS [*mask*]**

Lists all ISPF end-user interface views that can be captured. The *mask* option is a generic key that can be used to qualify the list.

**Note:** You must issue END or CANCEL to terminate a LIST task.

**LIST ALLOC**

Lists all the resources that have been allocated by the ALLOC command. Figure 29 on page 72 shows an example of the LIST ALLOC display.

## COD0 transaction

```

COD0 CICSplex/SM Debugger
CMD=>
Select P=Purge resource D=Dump resource
S Type Name Token Length MajObj
_ CACHE /@CACHE 01FF0004 000026E0
_ QUE /QUE A44C5E58 27257332 MAS
_ CLIST /C 000026E0 00106DF8 256
_ STG /STG 00000000 04289000 4096
_ EPTR /E 01FF0005 00001B00 2048
    
```

Figure 28. An example LIST ALLOC display

The fields on this display are:

### Field Description

**Type** The type of resource, as one of the following:

**EPTR** Data space pointer

**CLIST** Cache list

**CACHE**

Cache

**STG** CICS storage

**QUE** Queue ID

**Token** The 4-byte address or 8-byte token broken into two fullwords (ALET first).

### Length

The size of allocated storage or the element length for a cache list.

### MajObj

The major object, or component, used when allocating.

You can enter the following in the selection field:

### Command

#### Description

**P** Purges the resource.

**Note:** You cannot purge /@CACHE, which is the cache created by COD0.

**D** Enters CODB and dumps the resource. This is the same as entering DUMP /resname on the command line.

## LIST ATCB

LIST ATCB lists the API task control blocks. These control blocks are used when a CICSplex SM API operation is in progress.

The fields on this display are:

```

COD1 CICSplex/SM Debugger
CMD=>
D=Dump ATCB C=Dump CMDDesc T=Task END=Exit.
S ATCB Status Task# Cmd Origin DispTime APITime
_ 164B82B0 Active * 6337 GET_ IYEGZGW0/7186 00:00.00 00:32.93
_ 164B78D0 Avail * 6344
_ 164B5ED0 Free
_ 164B58D0 Free
_ 1649EED0 Free
_ 1649ECB0 Free
Total API Commands: 114,842
    
```

Figure 29. An example LIST ATCB display

#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#  
#

```

#           Field  Description
#
#           ATCB  The address of the ATCB control block
#
#           Status
#                   The status of the control block, as one of the following:
#
#                   Active  An API request is being processed.
#
#                   In Use
#                           An API request has been queued for processing.
#
#                   Avail  Available and waiting for work.
#
#                   Free   Available, but not waiting for work.
#
#           Task#  The CICS task number.
#
#           Cmd   The CICSplex SM API command being processed. This field is displayed
#                   only when Status=Active.
#
#           Origin For CICS based requests, Origin is in the form: aaaaaaaa/nnnnn where
#                   aaaaaaaa is the CICS APPLID and nnnnn is the CICS task number making
#                   the CICSplex SM API request being processed.
#
#                   For non-CICS based requests, Origin is the MVS jobname of the address
#                   space making the CICSplex SM API request being processed.
#
#                   This field is displayed only when Status=Active.
#
#           DispTime
#                   The elapsed time it took to dispatch the current API request.
#
#                   This field is displayed only when Status=Active.
#
#           APITime
#                   The elapsed time is has taken to process the current API request (Dispatch
#                   time not included).
#
#                   This field is displayed only when Status=Active.
#
#           The Total API Commands line displays the count of CICSplex SM API commands
#           that have been processed by all ATCBs.
#
#           Command
#                   Description
#
#           D      Dumps the ATCB
#
#           C      Dumps the API Command Descriptor if it is available.
#
#           T      Displays LIST TASK output for the CICS task processing the ATCB.
#
#           LIST CACHE
#           Lists the data caches in use by a local MAS. Figure 30 on page 74 shows an
#           example of the LIST CACHE display.
#
#           |
#           |
#           |

```

```

COD0 CICSPlex/SM Debugger
CMD=>
D=Dump lowest ALET:X'1000'
S Cache ALET Low Ofc High Ofc HWM Allocated (Hex)
_ DMDESCWW1 01FF001B 00000000 00801000 4,229,376 (00408900)
_ WLMICWW1 01FF0007 00000000 00400000 3,543,040 (00361000)
_ RTAICWW1 01010042 00000000 00400000 524,288 (00080000)
_ MONICWW1 01010041 00000000 00400000 524,288 (00080000)
_ TOPICWW1 01FF0006 00000000 00400000 655,360 (000A0000)
_ COMICWW1 01010040 00000000 00400000 1,691,648 (0019D000)
_ MASICWW1 01FF0008 00000000 00400000 933,888 (000E4000)
_ DATICWW1 0101003F 00000000 00400000 1,527,808 (00175000)
_ QUEICWW1 01FF0005 00000000 00400000 1,048,576 (00100000)
    
```

Figure 30. An example LIST CACHE display (CICS/ESA, CICS/MVS)

The fields on this display are:

**Field Description**

**Cache** The name of the data cache.

**ALET** The ALET of the data cache.

**Low Ofc**

The lowest offset allocated, which should always be X'00000000'.

**High Ofc**

The highest offset within the data cache allocated.

**HWM Allocated**

The number of bytes of the data cache actually in use.

**(Hex)** The HWM Allocated value expressed in hexadecimal.

Figure 31 shows an example of the LIST CACHE display.

The fields on this display are:

```

DBG0 CICSPlex/SM Debugger
CMD=>
Token ElemLen Keylen Keyoff Records MaxRecs FreeRec Storage
_ 00000001-0123FA2C 300 5 0 3 10 1 760
    
```

Figure 31. An example LIST CACHE display (CMAS)

**Field Description**

**Token** The cache list token.

**ElemLen**

The length of the element.

**Keylen**

The length of the key.

**Keyoff**

The offset of the key in each record.

**Records**

The number of records in the cache.

**MaxRecs**

The maximum number of records the cache can hold before being expanded.

**FreeRec**

The number of free slots available.

**Storage**

The total storage size, including any overhead.

You can enter the following in the selection field:

**Command**

**Description**

**D** Dumps the cache list data. This is the same as entering DUMP CACHE *cachetoken* on the command line.

**LIST CAPTURE**

Lists all ISPF end-user interface views currently being captured. Figure 32 shows an example of the LIST CAPTURE display.

The fields on the LIST CAPTURE display are:

```

COD0 CICSPlex/SM Debugger
CMD=>
Enter END to exit.
ViewName User ID Count Debug AltDebug Desc
FILES CVMCW2 2 0T NT File Summary
LOCFILE CVMCW2 2 08 N8 Local Files
    
```

Figure 32. An example LIST CAPTURE display

**Field Description**

**ViewName**

The name of the view for which data is being captured.

**User ID**

The TSO user ID from which the view command is being issued.

**Count**

The number of remaining times to capture the view.

**Debug**

The end-user interface program ID (that is, the selector) for the view being captured.

**AltDebug**

The back end program ID (that is, the action) for the view being captured.

**Desc**

A description of the view being captured.

**LIST CLIST**

Lists the data cache lists in use by the CMAS. Figure 33 shows an example of the LIST CLIST display.

The fields on this display are:

```

COD0 CICSPlex/SM Debugger
CMD=>
D=Dump the Cache List
S Token DataAlet DataStrt DataEnd EleSz ElemCnt Key Len T S Alt
_ 01FF001B00408990 01FF0007 00071000 00073C00 20 0 0 16 S B YES
_ 01FF001B00408830 01FF0007 0016E000 00177E5C 72 0 0 16 S B YES
_ 01FF001B004086D0 01FF0007 00165000 0016DCC4 64 0 0 16 S B YES
    
```

Figure 33. An example LIST CLIST display

**Field Description**

## COD0 transaction

**Token** The cache list token.

**DataAlet**

The ALET of the cache list's data.

**DataStrt**

The starting offset within the ALET allocated to the cache list.

**DataEnd**

The highest offset within the ALET allocated to the cache list.

**EleSz** The size of each cache list element.

**ElemCnt**

The number of elements in the cache list.

**Key** The offset of the key within an element.

**Len** The length of the key.

**T** The type of cache, as either standard (S) or generic (G).

**S** The search type for the cache, as either binary (B) or hash (H).

**Alt** Indicates whether there is an alternate index cache available.

### LIST COMM

Lists the two communication MAL queues: one for methods executing via the CMAS, and the other for methods routed to a MAS. Figure 34 shows an example of the LIST COMM display.

The fields on this display are:

```
COD0 CICSPlex/SM Debugger
CMD=>
S CSFM MAL      XLTD   Type   Node Type  Target  SysId Sequence
Response List for: MAL List
_ TSQO 001ABC00 00E1E300 Outbound Local MAS  CMAS1AB CAB1 00000012
Scroll=> PAGE
```

Figure 34. An example LIST COMM display

#### Field Description

**Response List for:**

**MAL List**

MAL execution to or from a CMAS.

**MAS List**

MAL execution to or from a MAS.

**CSFM** The method ID for the MAL being run.

**MAL** The address of the relocated MAL.

**XLTD** The address of the MAL descriptor table in the CMAS.

**Type** The type of communications in progress:

**Inbound**

The MAL is being run locally from another CMAS.

**Outbound**

The MAL is being sent to another CMAS.

**Response**

The MAL response is being transferred.

**Node Type**

The type of node involved in the transfer:

**CMAS** From a CMAS.

**Local MAS**

From a MAS in the same MVS image.

**LIST METH**

Lists all methods within the specified component. Figure 35 shows an example of the LIST METH display typical of those produced under CICS/ESA, and CICS/MVS. The fields on this display are:

```

COD0 CICSplex/SM Debugger
CMD=>
L=Dump Load Pt END=Exit.
S Typ Meth Function Fmt Tran LoadPt ServLevl Assembly Date Status
- PUB CWA ADDTMED 01 077130C0 CPSM210 05/19/98 06.05 ACTIVE
- PUB CWAB BROTMED 02 07713348 CPSM210 05/19/98 06.05 ACTIVE
- PUB CWAD DELTMED 03 077138F0 CPSM210 05/19/98 06.05 ACTIVE
- PUB CWAU UPDTMED 04 07713B40 CPSM210 05/19/98 06.05 ACTIVE
    
```

Figure 35. An example LIST METH display (CICS/ESA, CICS/MVS)

**Field Description**

**Typ** The type of method, as either public (PUB) or private (PRV).

**Meth** The method ID.

**Function**

The function name of the method.

**Fmt**

The format ID of the method.

**Tran**

If the method runs asynchronously, the CICS transaction ID used.

**LoadPt**

The load point of the method in memory.

**ServLevl**

The service level, or release level, of the method.

**Assembly Date**

The data and time at which the method was assembled.

**Status**

The status of the method as one of the following:

**ACTIVE**

The method is loaded.

**LOCK**

The method cannot be run locally. Either the method load detected errors or the method does not run in this environment.

**NOTFND**

The method is not in the load table for the specified release level of the CMAS or MAS.

**NOTRAN**

The transaction listed in the Tran field is not defined in the CICS PCT.

**TRAP1**

Trap level 1 is set for this method.

**TRAP1-2**

Trap levels 1 and 2 are set for this method.

**TRAP1-32**

Trap levels 1 – 32 are set for this method.

**LIST START**

Lists the status of all methods you've started or attached. Figure 36 shows an example of the LIST START display.

The fields on this display are:

```

COD0 CICSplex/SM Debugger
CMD=>
Select P=Purge V=View MAL D=Dump MAL END=Cont.
S Type Meth Task N Status
- START XQCQ 828 Completed, RESPONSE:OK
- START NSCR 844 Completed, RESPONSE:OK
- START NQPG 860 Completed, RESPONSE:EXCEPTION(ABEND)
    
```

Figure 36. An example LIST START display

**Field Description**

**Type** Either START or ATTACH, depending on which command you used to start the method.

**Meth** The name of the method.

**Task** The CICS task number of the method.

**Status**

The method's status as one of the following:

- Waiting for method to start or attach.
- Method is running.
- Completed, RESPONSE:<response>(<reason>).
- Method is no longer running!

**Note:** The error "Method is no longer running!" means the status in an internal table indicates the method should be running but the CICS task has been found not active via a CICS inquiry. This error is also used for attached tasks that may have timed out trying to communicate a request back into the CMAS.

You can enter the following in the selection field:

**Command**

**Description**

**P** Purges the MAL for this method.

**Note:** You cannot purge a MAL unless its status is "Completed".

**V** Formats the MAL.

**D** Calls the CODB transaction with the address of the MAL for hexadecimal dumps.

**A** Causes an ATTACH command to be created for the method with the context and scope of the original attach being viewed. The existing MAL is used as a starting point, but a new task will appear on the LIST START display.

**E** Causes an EXEC command to be created for the method with the context and scope of the original attach being viewed. Executed methods do not appear in the LIST TASK display; they are called directly by COD0 and the results are displayed immediately.

- S** Causes a START command to be created for the method with the context and scope of the original start being viewed. The existing MAL is used as a starting point, but a new task will appear on the LIST START display.

**LIST STCB**

Lists the server-client control blocks. These control blocks are used by CICSplex SM communications and the end-user interface to request work in a CMAS. Figure 37 shows an example of the LIST STCB display.

The fields on this display are:

```

COD0 CICSplex/SM Debugger
CMD=>
V=View MAL D=Dump MAL S=Dump STCB X=Dump XLSP
S Address Status Last Usr From Error CSFM Context Scope OutQue
_ 060C9BA0 Avail COM 0 TSPV PLEX2C1 PLEX2C1 NO
_ 060C97C0 Avail COM 0 TSCV NO
_ 060C93E0 Avail COM 0 CSAC NO
    
```

Figure 37. An example LIST STCB display

**Field Description**

**Address**

The address of the STCB control block.

**DataAlet**

The status of the control block as one of the following:

- Avail** Available and waiting for work.
- Free** Available, but not waiting for work.
- In Use** A MAL is being run.

**Timeout**

A conversation with the STCB timed out.

**Last Usr**

For a TSO end-user interface request, the user ID of the TSO user.

**From**

Where the request for this STCB came from, as either COM, for communications, or EUI, for the TSO end-user interface.

**CSFM**

The ID of the method last run using this STCB.

**Context**

The CMAS or CICSplex involved in the last request.

**Scope**

The CICSplex, CICS system group, or CICS system involved in the last request.

**OutQue**

For a TSO end-user interface request that was routed to multiple CMASs and MASs, indicates whether the status of each was reported individually or combined into a single response.

**LIST TASK**

Displays all CICSplex SM tasks and the methods being called within them. Figure 38 on page 80 shows an example of the LIST TASK display typical of those produced under CICS/ESA, and CICS/MVS.

|  
|  
|  
|  
|

## COD0 transaction

```

COD0 CICSPlex/SM Debugger
CMD=>
Scroll=> PAGE
DUMP L=Loadpt P=oPb O=Ossb S=Stack M=Mal B=modB E=moEb V=MAL END=Cont.
S Task # METH Load-pt oPb Ossb Stack Mal modB moEB
- 27 XLOP 00000000 00489FA8 00489FF0 0048A014 04283580 000CF820 00000000
- 27 DBG1 8A5B9690 00489FA8 00489FF0 0048A118 0A4602E4 000CF820 00000000
- 20 XLEV 00000000 00494FA8 00494FF0 00495014 04273580 000CF820 00000000
- 23 XLOP 00000000 00491FA8 00491FF0 00492014 04277580 000CF820 00000000
- 23 TIST 0A55C430 00491FA8 00491FF0 00492118 00490FD4 000D7D00 04274160
- 23 XSWC 0A514018 00491FA8 00491FF0 004925A0 0049244C 000D29F4 000DE0B0
- 24 XLOP 00000000 0048DFA8 0048DFF0 0048E014 0427F580 000CF820 00000000
- 24 RSWT 0A574728 0048DFA8 0048DFF0 0048E118 0048CFD4 000DAED4 042744D0
- 24 XSWC 0A514018 0048DFA8 0048DFF0 0048E4D8 0048E3BC 000D29F4 000DE0B0

```

Figure 38. An example LIST TASK display (CICS/ESA, CICS/MVS)

This display shows one line per method with a space between CICSPlex SM tasks. The fields on this display are:

### Heading

#### Description

#### Task #

The CICS task number.

**Note:** Do not use the task number to purge the CICSPlex SM transaction, as CICSPlex SM recovery will not be entered and CICSPlex SM system control block chains will be destroyed.

**METH** The name of the method running at that stack level.

#### Load-Pt

The address of the method's load point.

**oPb** The address of the object process block (one per CICSPlex SM CICS task) that points to all the OSSBs for this task.

**Ossb** The address of the stack segment block to which this method's stack is attached.

**Stack** The address of the method's stack.

**Mal** The address of the MAL for the method.

**modB** The address of the MODB for the component.

**moEB** The address of the MOEB for the component.

You can enter the following in the selection field:

### Command

#### Description

**L** Calls CODB to display the load point of the method.

**P** Calls CODB to display the OPB.

**O** Calls CODB to display the OSSB.

**S** Calls CODB to display the stack.

**M** Calls CODB to display the MAL.

- B** Calls CODB to display the MODB.
- E** Calls CODB to display the MOEB.
- V** Formats the MAL display as you would have if you entered it.
- U** Allows updating of the MAL in-flight.
- R** Lists the contents of all the registers (AR and GP). From this list you can enter:
  - D** Calls CODB to display data at that location using the AR register.
  - A** Calls CODB to display data at that location using only the general purpose register (ALET will be zero).

**CICSplex SM chain checking:** During a LIST TASK command the entire chain of CICSplex SM blocks that apply to a task are followed. The eyecatcher for each of the blocks is checked, in addition to the forward and backward methods within stacks and possible recursive chains. If any errors are found, you may see one of the following error messages after the last valid entry:

Stack chain broken at AAAAAAAAA

This error indicates that the previous method's ID within a stack chain does not match the previous method's ID. This may be the case if code within the method overlays the stack header. AAAAAAAAA is the address of the invalid stack frame.

OPB chain error at AAAAAAAAA

Object process blocks are created for each CICS CICSplex SM task. They are chained together for the component ID of the first method in the chain. If this chain points back to itself (a recursive chain), this message appears. AAAAAAAAA is the address of the OPB that was next after the previously displayed OPB.

Eyecatcher failed for CSFM at AAAAAAAAA

If an eyecatcher of a control block that is visited during a LIST TASK is incorrect, this message appears. AAAAAAAAA is the address of the control block in question and CSFM is its name.

DFHEIBLK block invalid at AAAAAAAAA, OPB at AAAAAAAAA  
invalid

The task's object process block is really the CICS DFHEISTG area. In this area is a pointer to the task's CICS EIB block, which is checked during LIST TASK commands.

## LIST VIEWS

Lists all ISPF end-user interface views that can be captured. You can qualify the list by using the *mask* option to specify a generic key. Figure 39 on page 82 shows an example of the LIST VIEWS display. The *mask* option was used to display only those views that begin with the letter A.

## COD0 transaction

```
COD0 CICSplex/SM Debugger
CMD=>
Enter END to exit.
View      DistName Type      Class Name      Desc
ACTNDEF   EYUZE GP0 TRTAA    ActDef          Action Definitions
ADMANLS   EYUZE AZ4 TMENU3   ViewTable       Real Time Analysis Admin Views
ADMCONFIG EYUZE AZ1 TMENU3   ViewTable       CMAS Configuration Admin Views
ADMIN     EYUZE $MM TMENU2   ViewTable       =====
ADMMON    EYUZE AZ2 TMENU3   ViewTable       Monitor Admin Views
ADMTOPOL  EYUZE AZ5 TMENU3   ViewTable       Topology Admin Views
ADMWRKLD  EYUZE AZ3 TMENU3   ViewTable       Workload Manager Admin Views
AIMODEL   EYUZE K50 TOPERATE CicsAiModel     Auto Install Models
AIMODELS  EYUZE K5S TOPERATE CicsAiModel     Auto Install Model Summary
ANALYSIS  EYUZE AZR TMENU1   ViewTable       Real Time Analysis Views
```

Figure 39. An example LIST VIEWS A\* display

The fields on this display are:

**Field Description**

**View** The name of the view.

**DistName**

The name of the source member for the view.

**Type** The type of view, which determines what selection list the view appears on.

**Class Name**

An internal description of the view.

**Desc** The description of the view as it appears on the end-user interface.

## POST (posting an ECB)

The POST command posts an ECB using the MVS POST command.

The format of the POST command is:

```
POST address
```

where *address* is a 1- to 8-character hexadecimal number that is the address at which the ECB resides.

No check is made to see whether an ECB exists at this address or whether it is already posted; an MVS POST command is simply issued.

**Note:** You can use the DUMP command or the CODB transaction to find the address.

## PRINT (printing data areas under CICS/ESA, CICS TS)

The PRINT command prints a CICSplex SM data area. PRINT uses the CICS spool facility to write the data area as an output file called Sxxxxxxx, where xxxxxx is a numeric identifier.

The format of the PRINT command is:

```
PRINT [parameters...]
```

where the parameters are:

```
/resname
```

Prints the specified allocated resource.

*alet addr size*

Prints an EPTR at the specified address for the specified number of bytes.

**CLIST** *token*

Prints the cache list of the specified token, where *token* is an 8-byte token entered as two 8-character hexadecimal fields.

**EIB** *compid*

Prints the EIB for the specified component.

**EIS** *compid*

Prints the EIS for the specified component.

*hexaddr size*

Prints memory at the specified address for the specified number of bytes.

**MAL** *addr*

Formats and prints the MAL at the specified address.

*method*

Prints the code for the specified method.

**MODB** *compid*

Prints the MODB for the specified component.

**MODD**

Prints the MODD.

**MOEB** *compid*

Prints the MOEB for the specified component.

**OPB** *compid*

Prints the object process block for the specified component.

**QUE** *token*

Prints the queue of the specified token, where *token* is an 8-byte token entered as two 8-character hexadecimal fields.

**STAKEND** *compid*

Prints the current stack for the specified component.

**STAKSTR** *compid*

Prints the first stack for the specified component.

**XLWA**

Prints the XLWA.

## PURGE (purging an allocated resource)

The PURGE command purges an allocated resource.

The format of the PURGE command is:

PURGE /resname

where:

*/resname*

Is the name of the resource you allocated. The storage assigned to the resource is removed from the system.

**Note:** You can also purge allocated storage using the P command from the LIST ALLOC display.

## START (starting a method in the CMAS)

The START command starts a method running within the CMAS.

The format of the START command is:

```
START method [termid]
```

where:

*method*

Is the name of a CICSplex SM method.

*termid*

Is a terminal ID.

The message argument list (MAL) of the method is displayed. For details about how to enter data from this display, see “Displaying a MAL from COD0” on page 86. For a list of valid responses to this command, see “Running a method” on page 89.

Figure 40 shows an example of the START display.

The START command starts a CICS task that eventually executes method DBG2.

```
COD0 CICSplex/SM Debugger
CMD=>                               Scroll=> PAGE
Overtpe fields and press ENTER to edit, END to proceed, CANCEL to abort.  Dn
IN
  ENM FUNCTION( SETCRGN )
  CHR DEBUG(          )
  BIN SYSTEM_AKP(          )
  BIN SYSTEM_AMAXTASKS(          )
  BIN SYSTEM_CUSHION(          )
  CHR SYSTEM_DTRPROGRAM(          )
  BIN SYSTEM_ECUSHION(          )
  BIN SYSTEM_MAXTASKS(          )
  BIN SYSTEM_MROBATCH(          )
  BIN SYSTEM_PRTYAGING(          )
  BIN SYSTEM_RUNAWAY(          )
  BIN SYSTEM_SCANDELAY(          )
  BIN SYSTEM_SYSDUMP(          )
  BIN SYSTEM_TIME(          )
  BIN TRACEDEST_AUXSTATUS(          )
  BIN TRACEDEST_GTFSTATUS(          )
  BIN TRACEDEST_INTSTATUS(          )
  BIN TRACEDEST_SWITCHSTAT(          )
  BIN TRACEDEST_SWITCHACT(          )
```

Figure 40. An example START display

This method is created dynamically by COD0 in every component.

## TRACE (setting CICS and CICSplex SM trace flags)

The TRACE command can be used to:

- Set CICS and CICSplex SM component trace flags
- Control output to auxiliary trace data sets

The format of the TRACE command is:

```
TRACE [parameters...]
```

where the parameters are :

**ON [RESET|START]**

**OFF** [RESET|STOP]

**USER** [RESET]

Controls the settings of the CICS component trace flags.

**ON** Turns all CICS component flags on, which produces slightly more output than the normal CICS trace settings.

**OFF** Turns all CICS component trace flags off, which results in almost no output at all (some CICS components do not have trace flags).

**USER** Traces only the CICS component application domains (AP0000 through APFFFF).

**RESET**

Causes tracing to start at the beginning of the auxiliary trace data set, overwriting any existing output.

**START**

Opens the auxiliary trace dataset.

**STOP** Closes the auxiliary trace dataset.

**FLAG**

Shows the trace flags of each CICSplex SM component. You can change the trace flag settings of one or more CICSplex SM components by overtyping the component's bit setting.

**SWITCH**

Switches the CICS auxiliary trace data sets and reports on which is active.

Changes made to CICS and CICSplex SM trace settings from the COD0 debugging transaction remain in effect after you exit the transaction.

## TRACK (setting trace flags by calling structure)

The TRACK command sets CICSplex SM trace flags based on the calling structure.

The format of the TRACK command is:

TRACK target relation calling flags id

where the parameters are:

*target*

The name of the method to be traced. You can provide a generic method name by specifying an asterisk (\*) at the end of the name or in place of the name (to indicate all methods).

*relation*

The relationship to the calling method as one of the following:

**FROM** Sets the trace for the target only when the direct caller is the calling method.

**STAK** Sets the trace for the target only if the calling method is somewhere in the CICSplex SM stack.

*calling*

The name of the method that calls the target method either directly or indirectly. You can provide a generic method name by specifying an asterisk (\*) at the end of the name or in place of the name (to indicate all methods).

## COD0 transaction

### *flags*

The trace flags to be set. The trace flags are set according to group names and are dependent on the underlying trace facility. The trace flags will be provided by IBM support should you need to use this facility.

**id** An optional user or task ID:

#### **Uxxxxxxx**

where xxxxxxxx is a 1-to 8-character EUI user ID.

#### **Tnnnnnnn**

where nnnnnnn is a 1- to 7-position CICS task number that can be obtained by issuing either the LIST TASK or CEMT INQ TASK command.

For example:

```
TRACK XD* STAK CI* SPEC UTOM
```

activates all trace flags for any data repository method that is called directly or indirectly from any communication initialization method, when the task is handling an EUI request from ID TOM.

## TRAP (setting trace flags for a method)

The TRAP command sets trace flags on for a specific CICSplex SM method.

The format of the TRAP command is:

```
TRAP method [1|2|ALL|OFF]
```

where:

#### **method**

Is the name of a CICSplex SM method.

#### **1|2|ALL|OFF**

Sets the trace flags for the specified method:

**1** Sets level 1 trace flags on.

**2** Sets level 1 and level 2 trace flags on.

**ALL** Sets level 1–32 trace flags on.

**OFF** Sets tracing for the method back to the flags specified on the COD0 TRACE command, the EYUPARMS start-up parameters, or the CMAS or MAS view command.

## Displaying a MAL from COD0

When entering into a MAL formatted by the COD0 debugging transaction, all input is validated for both physical and logical properties.

### **Format of the MAL display**

When viewing or updating a MAL, either from a START, ATTACH, or EXEC command, or from LIST output, the format of the display is as shown in Figure 41 on page 87.

```

COD0 CICSPlex/SM Debugger
CMD=>
Fields with "->" required.
IN
  *ENM FUNCTION( TEST )
  CHR DEBUG(      )
-> FLG FLAG_VALUES(      )
OUT
  *ENM RESPONSE(      )
  *ENM REASON(      )
    
```

Figure 41. Sample formatted MAL display

As shown in Figure 41, IN and OUT eyecatchers separate the major sections of the MAL. Each field name in the IN and OUT sections can be preceded by three other indicators:

- An arrow, indicating the field is mutually required or mutually exclusive and in error.
- An asterisk, indicating the existence bit for the field is set on in the MAL (OUT fields always have the existence bit on).
- A 3-character code indicating the field type.

**Note:** The FUNCTION field is completed by the COD0 debugging transaction and cannot be changed.

### Field types

The three-character code that precedes a field determines what can be entered in the field and the kind of data that is displayed. Table 5 shows the input allowed for each field type.

Table 5. Field types

Type	Format	Input allowed
BIN	BIN(n)	Hexadecimal number
BLK	BLOCK	Hexadecimal number for address or decimal number for length
BUF	BUFFER	Hexadecimal number for address, decimal number for length, or resource name (/resname)
CHR	CHAR(n)	Any character
CMP	COMPID	Component ID or '?' for a list
DEC	DEC(n)	Decimal number
EBK	EBLOCK	Hexadecimal number for ALET and OFFSET, decimal number for length, or resource name (/resname)
ENM	(names)	Names defined in format or '?' for list
EPT	EPTR	Hexadecimal number for ALET and OFFSET or resource name (/resname)
ETK	ETOKEN	Hexadecimal number or resource name (/resname)
FLG	FLAG	Hexadecimal representation of a flag or '?' for a list
LST	LIST	Hexadecimal number for address, decimal number for length, or resource name (/resname)
MPL	MAL	Hexadecimal number or resource name (/resname)
PTR	PTR	Hexadecimal number or resource name (/resname)
RES	RESTYPE	Resource name (/resname) or '?' for a list
SDT	SDT	TRUE or FALSE
STR	STRING(n)	Any character
TIM	TIMESTAMP	Hexadecimal number

Table 5. Field types (continued)

Type	Format	Input allowed
TKN	TOKEN	Hexadecimal number or resource name (/resname)

### Field edits and display formats

In a MAL display, input is edited and output is formatted according to the following rules:

#### Field type

##### Format

#### Hexadecimal

Hexadecimal characters 0–9 and A–F, in either upper or lower case. In output, the number is right-justified and padded with zeroes.

You can enter decimal characters instead of hexadecimal by preceding the value with a backslash, as in \1234. The decimal number is internally converted to hexadecimal.

#### Decimal

Numeric characters 0–9, without any sign. In output, the number is right-justified and padded with zeroes.

You can enter hexadecimal characters instead of decimal by preceding the value with a backslash, as in \ABCD. The hexadecimal number is internally converted to decimal.

#### ENM, CMP, or RES

One of the values shown in the message format. For example, if the format indicates MY\_FIELD IS (A,B,C,D), you can enter A, B, C, or D. You can also enter a question mark (?) to display a list of possible values; you can select one to be copied into the MAL.

For a field type of RES, you can use the HELP command to display a list of known resource types, such as HELP RESOP or HELP CVDA.

**Flags** A hexadecimal value representing a flag name, including a combination of flag names that have been logically ORed. You can also enter a question mark (?) to display a list of possible values. You can select as many as apply; they are logically ORed and copied into the MAL.

### Subfields

Many fields in a MAL consist of multiple subfields, which are divided into multiple input fields and validated separately. Each field is preceded by the suffix of the subfield. For example, the EPT field is made up of the ALET and OFFSET subfields, and looks like this when the MAL is displayed:

```
EPT YOUR_MAL_FIELD_NAME(A= alet O= offset)
```

The subfields associated with each field type are as follows:

#### Field type

##### Subfields

**EPT** A=alet, O=offset

**BUF** A=address, L=length, M=maximum length

**BLK** A=address, L=length

**EBK** A=alet, O=offset, L=length

**LST** A=address, N=number

### Using allocated resources

You can use the names of allocated resources (such as cache lists, data queues, data space storage, or CICS storage) in the input fields of a MAL. (For information on allocating resources, see “ALLOC (allocating a resource)” on page 63.)

If the field contains subfields, as described in “Subfields” on page 88, you need enter only the resource name in the first field; the COD0 debugging transaction determines the other field types and fills them in for you.

For example, you could use the ALLOC command to allocate 4K of data space storage to the resource called /workara, as shown in Figure 42.

Figure 43 shows the allocated resource, /workara, being used as input to a MAL.

```
COD0 CICSplex/SM Debugger
CMD=> ALLOC /workara EPTR 4096                               Scroll=> PAGE
IN
```

Figure 42. Using ALLOC to allocate a resource

The COD0 debugging transaction places the ALET of the allocated storage area

```
COD0 CICSplex/SM Debugger
CMD=>                                                         Scroll=> PAGE
IN
*ENM FUNCTION( TEST )
  CHR DEBUG(          )
  EPT OUT_ADDR(A= /workara  O=          )
```

Figure 43. Using an allocated resource in a MAL

into the A= field; the offset is automatically entered in the O= field.

### Running a method

From a formatted MAL display you can enter the following commands:

#### **CANCEL**

Returns control to the previous display without processing the MAL.

#### **DUMP** *mal-field*

Determines the type of the specified field and creates an appropriate DUMP command to call the CODB debugging transaction. *mal-field* can be any field on the formatted MAL display.

**END** Edits the MAL and then either ATTACHes, EXECutes, or STARTs the method. Control returns to the previous display.

If you return to the COD0 main menu while a method is running, a LIST START command is automatically issued.

#### **FLAG** *mal-field*

Displays a list of the specified field’s bit values (that is, their names from the Message Argument Format). Those that are currently set are prefixed by a plus sign (+). *mal-field* can be any field on the formatted MAL display that has a type of FLG. If the name you enter is neither part of the MAL nor an FLG field, an error message is issued.

**GO** Edits the MAL and then either ATTACHes, EXECutes, or STARTs the

## COD0 transaction

method. Control remains at the formatted MAL display. You can enter the same or different data, and issue GO or END again.

You can use the LIST or LIST START command to check the progress of the started or attached method. When you END the LIST display control returns to the formatted MAL display.

**NEXT** For the results of an ATTACHed method that either ran in multiple MASs or ran multiple times in a CMAS, displays the MAL that ran next.

**PREV** For the results of an ATTACHed method that either ran in multiple MASs or ran multiple times in a CMAS, displays the MAL that ran previously.

### Notes:

1. If you press Enter without issuing a command, the MAL is edited, but not run.
2. For the DUMP and FLAG commands, only fields that appear in the current formatted MAL display can be used as parameters. If you want to name the field of another MAL, you must first display that MAL from the LIST START or LIST TASK screen.

---

## System-level debugging with CODB

The CODB debugging transaction allows you to display and modify memory. It is menu-driven and allows you to choose various CICSplex data areas using PF keys or command line keywords.

After logging onto CICS, enter the CODB transaction ID to display the main menu, as shown in Figure 44. (CODB can also be entered from the DUMP command of the COD0 transaction.)

The first field is for the command, the second is for a component ID (which is

```
COMMAND==>          COMP ID==>      ADDR==>          ALET==> 00000000

      1. XLWA
      2. MODB
      3. MOEB
      4. OPB
      5. EIS
      6. EIB
      7. STAKSTRT
      8. STAKEND
      9. MODD
     10. MAL
     11. PFKON
     12. PFKOFF
     13. END
     14. CMASSTOP
     15. TRACE
     16. QUES
     17. MENU

P1=TOP P2=BOTM P3=END P4=PREV P5=NEXT P6=TOKEN P7=BACK P8=FRWD
P9=JUMP P10=DSJUMP P11=ALTER P12=ALET/OFFSET
MSG==>
```

Figure 44. CODB debugging transaction menu

required for some commands), the third is for the address (or AR mode offset), and the last is for an ALET or zeros.

**Note:** The CODB menu can be redisplayed at any time by issuing the MENU command.

## **CODB commands**

Any CODB command shown on the menu, or its associated number, is valid at any time. Some commands (such as MODB and MOEB) display a submenu listing the component ID and the address of the requested control block, if it can be located. The command name remains displayed until it is replaced by a new command, or a memory display is requested.

### **Command**

#### **Description**

**XLWA** Sets the ADDR==> field to the CMAS or MAS external linkage work area (XLWA) and the ALET==> field to zero, and displays the CICSplex anchor block.

### **MODB**

Displays the major object descriptor block (MODB) for the specified component.

**MOEB** Displays the major object environment block (MOEB) for the specified component.

**OPB** Displays the first object process block (OPB) for the specified component.

**EIS** Displays the CICS EXEC interface storage (EIS) block for the specified component.

**EIB** Displays the CICS EXEC interface block (EIB) for the specified component.

### **STAKSTRT**

Displays the first stack of the first transaction running for the specified component.

### **STAKEND**

Displays the current stack of the first transaction running for the specified component.

### **MODD**

Displays the major object director descriptor (MODD) block for the specified component.

**MAL** Displays the MAL currently initialized in the first transaction running for the specified component.

### **PFKON**

Turns on the PF key prompts at the bottom of the screen.

### **PFKOFF**

Turns off the PF key prompts at the bottom of the screen.

**END** Exits the CODB transaction.

### **CMASSTOP**

Shuts down the CMAS by posting the termination ECB.

### **TRACE**

Displays unformatted CICS internal trace table.

**QUES** Displays the queue token anchor block.

**MENU** Redispays CODB menu.

|  
|  
|  
|

## CODB transaction

**Note:** The MODB, MOEB, OPB, EIS, EIB, STAKSTRT, STAKEND, MODD, and MAL commands require a component ID, as described in “The COMP ID field.”

The MENU command can be issued at any time to redisplay the CODB menu.

## The COMP ID field

CODB commands that display CICSplex SM control blocks (such as MODB and MOEB) require you to specify a three-character component ID in the COMP ID field. For a list of valid component IDs, see “Major components of CICSplex SM” on page 139.

When you have specified a component ID, it remains displayed until one of the following occurs:

- A new component ID is specified.
- The COMP ID field is blanked out.
- A memory display is requested.
- A command is entered that does not require a component ID.

So it is possible to display various control blocks belonging to a single component by establishing the component ID and then issuing different commands.

## The ADDR field

Entering a value in the ADDR field produces a display of memory at the specified address, using the current ALET. If the address cannot be accessed, a message appears in the MSG field at the bottom of the display.

Relative addressing is also supported in the ADDR field. You can enter a scroll amount, in bytes, as a signed (+ or –) hexadecimal number. For example:

```
ADDR==> +2D0
```

## The ALET field

Entering a value in the ALET field sets the ALET value to be used for memory displays. This field is normally filled in; it has an initial value of hexadecimal zeros.

## The PF Key prompts

The PF key prompt area contains a two-line list of the PF keys supported and a brief description of their values. This prompt can be turned off by the PFKOFF (12) command and turned back on by the PFKON (11) command.

The following PF keys are in effect while the CODB transaction is running:

Key	Description
PF1	TOP (valid only for control block displays). Repositions the display to the beginning of the control block. If the display was produced by a value in the ADDR field, a warning message appears in the MSG field.
PF2	BOTTOM (valid only for control block displays). Repositions the display to the end of the control block. If the display was produced by a value in the ADDR field, a warning message appears in the MSG field.
PF3	END. Exits the CODB transaction.
PF4	PREV. Depending on the contents of the current display, displays the previous control block of the same type or the previous cache list or queue record.

For a control block display, PREV is both command (control block) and component sensitive. If a submenu from a control block command is displayed, PREV displays the last component's control block, if it exists; if it does not exist, a warning message appears in the MSG field. If a component's control block is displayed, the previous component's control block is displayed.

For a cache list or queue record display, if you issue PREV when the first record is displayed, a warning message appears in the MSG field.

**PF5** NEXT. Depending on the contents of the current display, displays the next control block of the same type or the next cache list or queue record.

For a control block display, NEXT is both command (control block) and component sensitive. If a submenu from a control block command is displayed, NEXT displays the Kernel Linkage control block, if it exists; if it does not exist, a warning message appears in the MSG field. If a component's control block is displayed, the next component's control block is displayed. When displaying OPBs, NEXT runs down each component's OPB chain, if it exists, before going on to the next component.

For a cache list or queue record display, if you issue NEXT when the last record is displayed, a warning message appears in the MSG field.

**PF6** TOKEN. Displays either the first record of the queue whose QTOKEN is pointed to by the cursor, or the first cache list element whose EPOINTER is pointed to by the cursor. The NEXT and PREV commands can be used to scroll forward and backward through the queue or cache list.

**PF7** BACKWARD. Scrolls the memory display backward one full page.

**PF8** FORWARD. Scrolls the memory display forward one full page.

**PF9** JUMP. Produces a display that starts at the address pointed to by the cursor, using an ALET of zero. The address pointed to can be the address field, the relative address field, the EBCDIC field, or an address in the hexadecimal data display. If the specified memory cannot be accessed, a warning message appears in the MSG field.

If a control block was being displayed, JUMP erases the current command and component ID and establishes the ADDR mode. After a JUMP command, it is possible to scroll beyond the bounds of the control block, even if the address selected is within the block. To reestablish control block mode, the desired command and component ID must be reentered.

**PF10** DSJUMP. Produces a display that starts at the address pointed to by the cursor, using the displayed ALET. The address pointed to can be the address field, the relative address field, the EBCDIC field, or an address in the hexadecimal data display. If the specified memory cannot be accessed, a warning message appears in the MSG field.

**PF11** ALTER. Allows you to alter memory.

**PF12** ALET/OFFSET. Produces a display that starts at the ALET/ADDRESS pair pointed to by the cursor. The ALET/ADDRESS pair must be in the hexadecimal data display and the cursor must be on the ALET portion of the pair. If the specified memory cannot be accessed, a warning message appears in the MSG field.

## COBD transaction

## The MSG field

This is a one line area headed by: MSG==> that appears on all screens. The MSG field is used for warning, informational, and error messages. For a list of these messages and their meanings, refer to the *CICSplex System Manager Messages and Codes*.

## The memory display area

The memory display area contains hexadecimal and EBCDIC representations of the requested memory ALET/ADDRESS, or the requested control block. Each line of the display contains an address, its offset from the beginning of the area (either the start of the control block or the address entered in the ADDR field), four full words of data in hexadecimal format, and the EBCDIC representation of those sixteen bytes. Figure 45 is a sample COBD memory display.

```
COMMAND==> XLWA      COMP ID==>      ADDR==>      ALET==> 00000000
00077368 00000000 020C6EC5 E8E4E7D3 D2D5D3C3 E6C1C1C2 ..>EYUXLKNLCWAAB
00077378 00000010 01000200 000773E0 00000000 006B2F20 .....\......
00077388 00000020 00000000 0000000E 0000E888 00097820 .....Yh....
00077398 00000030 00077470 0000A81C 0008D000 000003B6 .....y...}....
000773A8 00000040 00084E90 00040000 043E0000 00000020 ..+......
000773B8 00000050 043E0000 FFFF34E 8A680940 006B2F20 .....3+... ,..
000773C8 00000060 00000000 D2D3D7C2 00085310 00085828 ....KLPB.....
000773D8 00000070 0A62AD40 0093D154 00077384 000773C4 ... .lJ...d...D
000773E8 00000080 00000000 00000000 00000000 00346EC5 .....>E
000773F8 00000090 E8E4E7C5 C5E8E4D9 E7C5D3E2 01030000 YUXEYURXELS...
00077408 000000A0 009AFC38 03C6B150 0093D140 0093D154 ....F.&;lJ .lJ.
00077418 000000B0 00FAB580 006C1258 A458C562 09D66631 .....%.u.E..0..
00077428 000000C0 D7D9D4C2 000774B8 00077528 0007752C PRMB.....
00077438 000000D0 00077530 00077534 00077548 00077544 .....
00077448 000000E0 00077558 0007755C 0007756C 00077560 .....*...%...-
00077458 000000F0 00077568 00077564 00077570 00000000 .....
00077468 00000100 00000000 00000000 00000000 00097820 .....
00077478 00000110 000988BC 00099958 0009A9F4 0009BA90 ..h...r...z4...
00077488 00000120 0009CB2C 0009DBC8 0009EC64 0009FD00 .....H.....
00077498 00000130 000A0D9C 000A1E38 000A2ED4 000A3F70 .....M....
000774A8 00000140 000A500C 00000000 00000000 00000000 ..&;.....
000774B8 00000150 A458C562 09D66631 006C1258 00FAB580 u.E..0...%.....
000774C8 00000160 000000D4 00000002 E2E8E2C3 C3E5D4C3 ...M...SYSCVMC
000774D8 00000170 E3E2D6F1 C3E5D4C3 E6404040 C3E6E6F1 TS01CVMCW CWW1
000774E8 00000180 DF80FCA0 00800000 00000000 00000000 .....
000774F8 00000190 00000000 00000000 00000000 00000002 .....
00077508 000001A0 04375000 00000000 00000000 00000000 ..&;.....
00077518 000001B0 00000000 00000000 00000000 00000000 .....
00077528 000001C0 00000000 00000000 00000000 00000400 .....
00077538 000001D0 00040000 00000002 0000001C 00000000 .....
00077548 000001E0 C5E8E4C4 D9C5D740 00000000 00000000 EYUDREP .....
00077558 000001F0 00000000 00000000 00000000 00000000 .....
00077568 00000200 00000000 00000000 00000000 .....
```

P1=TOP P2=B0TM P3=END P4=PREV P5=NEXT P6=TOKEN P7=BACK P8=FRWD  
P9=JUMP P10=DSJUMP P11=ALTER P12=ALET/OFFSET

Figure 45. Sample COBD memory display

## COBD altering memory

The hexadecimal and EBCDIC data portions of the display can be modified. You can overtype hexadecimal data using valid hexadecimal digits, or EBCDIC data using any keyboard character except the period. After overtyping the data, press PF11 (ALTER).

**Note:** The CODB alter memory function should be used only at the request of customer support personnel.

A warning message appears in the MSG field if:

- The memory is protected.
- You altered the screen but did not press PF11.
- The memory location being altered has changed since it was displayed.

Trying to modify protected storage causes an abend. The CODB recovery routine issues a message describing the abend to the console.

### Accessing CODB from COD0

CODB can be entered from the COD0 transaction by:

- Using the DUMP command.
- Entering a D in a selection field, when allowed.

When you exit CODB (by issuing the END command) you are returned to the COD0 transaction.

There are some advantages to using COD0 to enter CODB:

- The DUMP command translates a method name into its entry addresses so you can dump or alter method code.
- From the LIST TASK screen you can dump individual stacks, MALs, OPBs, and OSSBs, for example.
- You can dump allocated resources (as defined by the ALLOC command) by name, and COD0 translates them into ALET/OFFSETS, ADDRESSES, or TOKENS, as required.
- You do not need to know the exact ALET/OFFSET or ADDRESS of the area you are dumping.

## CODB transaction

---

## Chapter 11. Using PlexManager diagnostic facilities

This chapter describes how to use the following PlexManager diagnostic facilities:

- The DIAGMSG view
- The DIAGSESS view
- The DIAGSYS view
- Extended diagnostic mode (XDM)

### Usage Note

The DIAGMSG view and extended diagnostic mode (XDM) should be used only at the request of customer support personnel.

---

### The DIAGxxxx views

This section describes how to use:

- The DIAGMSG view, to display and update the current status of PlexManager diagnostic facilities.
- The DIAGSESS view, to display information about active CAS-to-CAS communication sessions. (Prior to Release 2 of CICSplex SM, the DIAGSESS view was called the SESSIONS view.)
- The DIAGSYS view, to display information about CASs connected to the local CAS. (Prior to Release 2 of CICSplex SM, the DIAGSYS view was called the SYSTEMS view.)

### DIAGMSG (Diagnostic facilities)

The DIAGMSG view displays the current status of the PlexManager diagnostic facilities. By default, the diagnostic facilities are disabled (their status is OFF). They should be enabled *only* at the request of customer support personnel. From the DIAGMSG view, you can enable or disable all of the diagnostic facilities except GXDM and LXDM (see the Note in Table 6 on page 98 for details).

To display the DIAGMSG view from any PlexManager or CICSplex SM view, issue the command:

```
DIAGMSG [msgoption]
```

where:

*msgoption*

Is the specific or generic name of a diagnostic message option. Valid options are:

**msgoption**

**Description**

**GEMM**

Global extended message mode

**GESTR**

Global extended security trace

**GSSM** Global safe security message

**GSSTR**

Global simple security trace

**GXDM** Global extended diagnostic mode

**LEMM** Local extended message mode

## DIAGMSG

**LESTR** Local extended security trace  
**LSEMM** Local security extended message mode  
**LSSTR** Local simple security trace  
**LXDM** Local extended diagnostic mode  
**WSXASTR** Window security extended authorization simple trace

If you specify no *msgoption* value, the view includes information about all diagnostic message options available at your enterprise, as illustrated in Figure 46.

```
27FEB2005 16:19:59 ----- INFORMATION DISPLAY -----
COMMAND ==>
CURR WIN ==> 1          ALT WIN ==>
W1 =DIAGMSG=====EYUA=====27FEB2005==16:19:59=PLEXMGR=====11===
CMD Option  Status Scope  Description / Diagnostic Activity
-----
GXDM      OFF   Global Extended Diagnostic Mode
LXDM      OFF   Local  Extended Diagnostic Mode
GEMM      OFF   Global Extended Message Mode
LEMM      OFF   Local  Extended Message Mode
LSEMM     OFF   Local  Security Extended Message Mode
LESTR     OFF   Local  Extended Security Trace
GESTR     OFF   Global Extended Security Trace
LSSTR     OFF   Local  Simple Security Trace
GSSTR     OFF   Global Simple Security Trace
GSSM      OFF   Global Safe Security Message Display
WSXASTR   OFF   Window Extended Authorization Simple Trace
```

Figure 46. The DIAGMSG view

### Action commands

Table 6 lists the action commands for the DIAGMSG view.

Table 6. DIAGMSG action commands

Primary command	Line command	Description
n/a	ON	Enables the specified diagnostic message option.
n/a	OFF	Disables the specified diagnostic message option.

**Note:** The DIAGMSG action commands are not valid for GXDM or LXDM, which must be enabled and disabled either by updating the start up procedure for the CAS or CMAS (see *CICS Transaction Server for z/OS Installation Guide*) or by updating the XDM mode field on the Session Control Parameters panel displayed when you select CICSplex SM option 0 Profile and then option 1 Control.

### Hyperlink fields

None.

## DIAGSESS (Connected sessions)

The DIAGSESS view displays information about the communication sessions that are currently active between the local CAS and other CASs directly connected to it. Such information can help you assess the impact of stopping any VTAM® communication link.

To display the DIAGSESS view, you can:

- From any PlexManager view, issue the command:  
DIAGSESS [session]

where:

*session*

Is a unique identifier assigned to each session. If you specify no session identifier, information about all known sessions is displayed, or

- From a CICSplex SM window, issue the CONTEXT display command, as shown here:

```
CONTEXT * PLEXMGR; DIAGSESS
```

where:

**\* (asterisk)**

Is the CAS you are currently signed on to. If you prefer, you can replace the asterisk with the MVS sysid of any CAS connected to your local CAS and running PlexManager.

**PLEXMGR**

Is the product context you require.

**DIAGSESS**

Is the name of the PlexManager view you want to display.

**Note:** If the PlexManager DIAGMSG option LSEMM is enabled, message BBMSS2011 is displayed each time you open a new window or change the context of a window. To remove the message, press Enter.

OR

- From a CICSplex SM window, issue the SET display command to display the SET WINDOW CONTEXT, PRODUCT, SCOPE and VIEW input panel. Complete the panel as shown here, and press Enter:

```
-----SET WINDOW CONTEXT, PRODUCT, SERVER, SCOPE AND VIEW -----
COMMAND ==>                                     ==> PAGE

Window Parameters:

Context      ==> *
Product      ==> PLEXMGR
Server       ==> *
Scope        ==> *
View         ==> DIAGSESS

Type End to Set Window Parameters
Cancel to quit without setting
```

Figure 47. Using SET to display a PlexManager view

The asterisk (\*) in the Context field identifies the CAS you are currently signed on to. If you prefer, you can replace the asterisk with the MVS sysid of any CAS connected to your local CAS and running PlexManager.

Figure 48 on page 100 shows an example of the DIAGSESS view.

## DIAGSESS

```
27FEB2005 16:19:45 ----- INFORMATION DISPLAY -----
COMMAND ===>                                SCROLL ===> PAGE
CURR WIN ===> 1          ALT WIN ===>
W1 =DIAGSESS=====EYUA=====27FEB2005==16:19:45=PLEXMGR=====4===
CMD Sess Source          Destination      ConvStrt Send  Send  Recv  Recv
--- ID   System/User      System/User      ----- Req  Data  Req   Data
      46 *.PLEXMGR        EYUA.DAVEJEF    16:18:20  34   54   30   16
      42 *.DAVEJEF        EYUA.PLEXMGR    16:18:20  31   16   34   54
      22 *.PLEXMGR        EYUB.EYUJCSS2   15:32:07  101  128  99   20
      17 *.EYUJCSS1       EYUB.PLEXMGR    15:31:41  100  20   102  149
```

Figure 48. The DIAGSESS view

### Action commands

There are no action commands for the DIAGSESS view.

### Hyperlink fields

There are no hyperlink fields in the DIAGSESS view.

## DIAGSYS (Connected CASs)

The DIAGSYS view displays information about the CASs that are directly connected to the local CAS. To display the DIAGSYS view, you can:

- From any PlexManager view, issue the command:

```
DIAGSYS [system]
```

where:

*system*

Is the 1- to 8-character name of a connected system. If you specify no system value, information about all connected systems is displayed.

OR

- From a CICSplex SM window, issue the CONTEXT display command or the SET display command, as described for the DIAGSESS view on page 99, replacing DIAGSESS with DIAGSYS.

Figure 49 is an example of the DIAGSYS view.

```
27FEB2005 16:19:59 ----- INFORMATION DISPLAY -----
COMMAND ===>                                SCROLL ===> PAGE
CURR WIN ===> 1          ALT WIN ===>
W1 =DIAGSYS=====EYUA=====27FEB2005==16:19:59=PLEXMGR=====2===
CMD SYSTEM  C SMF  SUBS  APPLICAT  MODE  ACTSESS
-----
EYUA      *      EYUA  EYUCAS1A      2
EYUB      *      EYUB  EYUCAS1B      2
```

Figure 49. The DIAGSYS view

### Action commands

There are no action commands for the DIAGSYS view.

### Hyperlink fields

There are no hyperlink fields in the DIAGSYS view.

## Extended diagnostic mode

The PlexManager extended diagnostic mode (XDM) can be used to collect additional diagnostic data for problems found in the CAS, CMAS, or TSO user address space. Specifically, XDM provides:

- Extensive status messages
- Extended internal tracing
- Disabling of critical service error recovery

To activate XDM mode in the CAS or CMAS, update their respective start-up procedures by adding XDM=Y on the EXEC statement. To activate XDM mode in the TSO address space, specify XDM=YES when the CAS Connection ISPF Panel is displayed.

## Status messages

When XDM is active, the following messages can be issued to the console:

**BBMSD801I Module - xxxxxxxx not found. Conditional PGLoad request failed**

This message is issued whenever a conditional MSD\_PGLoad request fails.

**BBMSD900E TCB AT xxxxxxxx HAS ABNORMALLY TERMINATED - CODE=tnn**

This message is issued any time a task abends.

**BBMZE989I CONTROL TASK (nnnn) RESOURCE RECOVERY COMPLETE - jobname(jesjobid)**

This message is issued whenever a control task for any connected address space goes through end of task processing.

**BBMZE999I GLOBAL RESOURCE RECOVERY COMPLETE - jobname(jesjobid)**

This message is issued whenever any connected address space goes through end of memory processing.

In addition to these extra diagnostic messages, all abends are documented with LOGREC, SDUMP, and recovery manager messages.

If the General Services Interface (GSI) router returns a failure code with a level higher than WARNING and the GSI requester is not a local server initialization function, a snap trace and dump of the failing CSR and FPB are produced. The snap trace and dump consist of messages like these:

```
BBMLD001E GSI service failed - Completion code=8
BBMLD002E GSI call issued at 85112B96 (MXC42+0452 in control section MXC40)
BBMLD003E CSR at 7F418F7C; ROUT=01010415 / ZERR=07D33212 / FPBA=7F4D017C
BBMLD004E FPB data 0000 0000007C 00000038 15000000 7F455AF0
BBMLD004E ..... 0010 00000008 07D33212 07080200 7F4D017C
BBMLD004E ..... 0020 00000000 7F455398 01010200 01049C00
BBMLD004E ..... 0030 00000000 00000000 00000000 7F455AF0
```

## Internal tracing

Enabling XDM activates the internal linkage trace in a CAS. The linkage trace is kept in ECSA in 4K blocks. It keeps track of both assembler and C services:

- Prolog (Module Entry)
- Epilogue (Module Exit)
- Language Environment® Create
- Language Environment Clone
- Language Environment Destroy

## extended diagnostic mode

If XDM is active, the linkage trace also causes the creation of storage manager and environment performance instrumentation GTF records. These GTF records help in determining performance-related problems.

**Note:** To activate fully the creation of these GTF records, SMGTF=Y must also be specified in the CAS startup procedure.

## Disabling error recovery

During normal system operation, CICSplex SM does everything it can to prevent critical services, such as the storage manager and the program manager, from terminating. However, when XDM mode is active, these services abend in cases that would normally produce a bad return code.

For example, the storage manager normally passes back a warning completion code in the following conditions:

- Invalid parameter list
- Invalid registered block ID
- Requester is not authorized
- Unable to access the storage block
- Abend during processing

With XDM active, all of these conditions result in an error completion code. Also, if an error occurs during component initialization or termination, the controller is terminated.

---

## **Part 3. Investigating and documenting a problem**

This part of the book describes troubleshooting techniques for determining the cause of a CICSplex SM problem. It may help you solve some problems yourself. It will also guide you in collecting the necessary documentation for problems that you cannot solve yourself. Such problems should be referred to your IBM Support Center.

Each chapter in this part discusses different types of problem that you may have.



---

## Chapter 12. Abends, stalls, and bottlenecks

The following sections present information to help you troubleshoot several types of problems:

- “Investigating abends”
- “Investigating stalls” on page 106
- “Investigating bottlenecks” on page 107

---

### Investigating abends

Because CICSplex SM has a presence in two major parts of your environment (MVS and CICS) abends can occur in either place. CAS, ISPF end-user interface, and some CMAS abends occur under MVS. MAS and other CMAS abends, however, occur under CICS.

Use the information in this section to help you isolate the cause of an abend or to report the condition to customer support personnel.

### MVS abends

#### What CICSplex SM does

- Passes control to the appropriate recovery routine.
- Produces an SDUMP.
- Writes BBx and EYU messages to the console, job log, and EYULOG.

If the CAS abends, your CMASs and MASs are not affected. However, you do lose the ability to access CICSplex SM through the ISPF end-user interface.

#### Documentation to collect

- System console log and EYULOG
- CAS job log
- Unformatted SDUMP from the affected address space
- AUXTRACE data set, if available
- Any LOGREC entries

An unformatted SDUMP is the preferred source of problem diagnosis information for an abend. You should format a CICSplex SM dump only at the request of customer support personnel.

### CICS abends

#### What CICSplex SM does

- Passes control to CICS, which decides whether to take an SDUMP.
- Regains control from CICS.
- Produces a transaction dump and, possibly, an SDUMP.
- Writes an EYU failure summary to the console.
- Writes EYU messages to the job log and EYULOG.

#### Documentation to collect

- System console log and EYULOG
- Unformatted SDUMP from the affected address space
- AUXTRACE data set, if available

| An unformatted SDUMP is the preferred source of problem diagnosis information for  
| an abend. You should format a CICSplex SM dump only at the request of customer  
| support personnel.

---

## Investigating stalls

When CICSplex SM doesn't seem to be responding, you should suspect a stall condition, which could be either a loop or a wait.

**Note:** In the case of a suspected loop or wait, you should request an SDUMP; CICSplex SM will not take one automatically. However, do not cancel the task that appears to be stalled before requesting the dump. If you cancel the task, the CICS and CICSplex SM recovery routines that get control will change the "picture" taken by the dump and you may lose valuable diagnosis information.

You will need to determine both at what stage of processing the stall occurred and where it occurred. Processing a CICSplex SM request involves multiple address spaces. The process could stall in the TSO/ISPF session, in any of the CICS systems included in the current context and scope, or at any of several points in between.

Use the information in this section to help you isolate the cause of a stall or to report the condition to customer support personnel.

## An undetermined stall condition

### Questions to ask

- Did the stall occur during initialization?
  - How far did initialization progress?
  - Were there any definition or setup errors reported?
- Did the stall occur during operation?
  - Are the necessary communication links between CASs, CMASs, and MASs available?
  - What type of request was being processed?
  - How big was the CICSplex involved?
  - How many CMASs and MASs were involved?
  - What types of monitoring, real-time analysis, and workload management were active?
- Did the stall occur during termination?
- Did the stall occur in the ISPF end-user interface?
- Did the stall occur in a CAS?
- Did the stall occur in a CMAS?
  - Did the request time out with an EYUEInnnn message?

The local CMAS may be waiting for one or more CICS systems (or their CMAS) to return requested data. A CICSplex SM view does not return until all the expected data is collected.
  - Did the request time out with a CICS message?
- Did the stall occur in a MAS?

Try stopping the MAS agent code (using the STOP action command from the MAS view), then evaluate the underlying CICS system.

  - Is the CICS system taking an SDUMP?
  - Is the CICS system looping or hung?
  - Did the request time out with a CICS message?

- Is the CICS system experiencing a short on storage (SOS) condition, or has it reached its MAXTASK level?

Any one of these conditions could prevent some types of CICSplex SM requests from completing.

#### Documentation to collect

- System console log and EYULOG
- CAS and CMAS job logs
- Unformatted SDUMP from the affected address spaces (TSO, CAS, CMAS or MAS)

## A suspected loop

#### Questions to ask

- What are some possible sources of the loop?
- Is CPU usage particularly high?

#### Documentation to collect

- Appropriate job logs
- Selected trace data, as requested by support
- AUXTRACE data set, if available
- Transaction dump, if any
- CICS system dump, if any

## A suspected wait

#### Questions to ask

- At what point is the wait occurring?
- Is CPU usage particularly low?

#### Documentation to collect

- Appropriate job logs
- Appropriate CICS CEMT queries
- Selected trace data, as requested by support
- AUXTRACE data set, if available
- Transaction dump, if any
- CICS system dump, if any

An unformatted dump is the preferred source of problem diagnosis information for a stall. You should format a CICSplex SM dump only at the request of customer support personnel.

---

## Investigating bottlenecks

Bottlenecks can be caused by various components of CICSplex SM. You need to be aware of how these components are defined and how they interact, as well as of the transactions underway when the bottleneck occurs.

Use the information in this section to help you isolate the cause of a bottleneck or to report the condition to customer support personnel.

#### Questions to ask

- What type of request was being processed?
- How big was the CICSplex involved?
- How many CMASs and MASs were involved?

## bottlenecks

- What types of monitoring, real-time analysis, and workload management were active?
- What are the dispatching priorities of the CMASs and MASs?  
The priority of a CMAS must be higher than that of the MASs it manages.
- Are the CICS SIT parameters correctly specified for the CMASs and MASs?
- How is the communications network performing?

### **Documentation to collect**

To diagnose a performance problem such as a bottleneck, customer support personnel may ask you to turn on trace level 16 in selected CICSplex SM components. Many components use trace level 16 to determine how long a request takes to complete. It may be possible, based on that data, to isolate the problem to outgoing or incoming processes. For information on controlling the trace levels in CICSplex SM components, see “Controlling the amount of tracing in a CMAS or MAS” on page 29.

---

## Chapter 13. Investigating output and system management problems

This chapter describes some ways of solving typical problems with output and system management results.

When you have problems with unexpected or incorrect output from the end-user interface, customer support personnel may ask you to provide screen prints showing the problem, in addition to other types of documentation listed in Chapter 12, “Abends, stalls, and bottlenecks,” on page 105.

---

### Incomplete operations data returned

Consider this example of incomplete data that is returned to the end-user interface in response to an operations view command:

```
27FEB2005 07:02:28 ----- INFORMATION DISPLAY -----  
COMMAND ===>                                     SCROLL ===>  
CURR WIN ===> 1           ALT WIN ===>  
W1 =CICSRGN=====EYUPLX01=EYUCSG01=27FEB2005==07:01:51=CPSM=====  
BBMXBD15I There is no data that satisfies your request
```

Figure 50. Example of incomplete data returned

In the example shown in Figure 50, there is a CICS system known as EYUMAS1A. EYUMAS1A has been installed as a MAS. It is currently running, yet it does not show up as expected on the CICSRGN view.

A good first step to determine what is wrong is to issue the MAS view command, using the same context (EYUPLX01) and scope (EYUCSG01) as the failing CICSRGN view. The result of issuing this command is one of the following:

- There is no entry for EYUMAS1A.
- The entry for EYUMAS1A shows a status of INACTIVE.
- The entry for EYUMAS1A shows a status of ACTIVE.

### No entry in the MAS view

Three possible causes for there being no entry for EYUMAS1A in the MAS view are:

1. The scope EYUCSG01 is incorrect.  
If EYUMAS1A is not a member of the CICS system group EYUCSG01, the scope is incorrect. To test that possibility, use the SCOPE command to change the scope to either the entire CICSplex (EYUPLX01) or the MAS itself (EYUMAS1A). If the refreshed MAS view with the new scope shows an entry for EYUMAS1A, the problem was an incorrect scope.
2. The context EYUPLX01 is incorrect.  
EYUPLX01 should have been the context when the CICSSYS definition for EYUMAS1A was created. If it was not, use the CONTEXT command to refresh the MAS view, using the correct context.
3. The CMAS was initialized with the wrong data repository.

## incomplete operations data returned

### INACTIVE status

Whenever either a MAS or a CMAS is started, CICSplex SM attempts to activate communication between the MAS and the CMAS. If both the CMAS and the MAS are running and the status on the MAS view shows INACTIVE, you need to look at the JESMSG LG of the MAS and the EYULOG of the CMAS. They may contain messages indicating that the connection process failed and suggesting what could be wrong.

It could be that the CICS SYS definition name does not match the EYUPARM parameter NAME in the startup JCL for the MAS. It is also a possibility that, if the default for the EYUPARM NAME is taken, EYUMAS1A is not the VTAM APPLID. Here is an example of the JESMSG LG of the MAS when the NAME parameter is incorrect:

```
DFHSI1517 EYUMAS1A Control is being given to CICS.
EYUXL0003I EYUMAS1A CPSM Version 210 LMAS startup in progress
EYUXL0022I EYUMAS1A LMAS Phase I initialization complete
EYUXL0020I EYUMAS1A ESSS connection in progress to CICSplex(EYUPLX01)
EYUXL0004I EYUMAS1A ESSS connection complete
EYUCL0112E EYUMAS1A Protocol Services initialization unable to perform ICT Attach
EYUCL0101E EYUMAS1A Protocol Services initialization failed
EYUCI0101E EYUMAS1A Communications initialization failed
EYUXL0112E EYUMAS1A LMAS initialization failed
```

*Figure 51. Example of JESMSG LG when EYUPARM NAME parameter is incorrect*

The EYUPARM parameter CICSplex in the startup JCL for the MAS may not match the CICSplex name being used as the context for the MAS view. If the CICSplex named in the EYUPARM is valid, the MAS probably connected successfully to that CICSplex, instead of to the CICSplex used as the context for the MAS view that shows INACTIVE.

If SEC(NO) is coded in the EYUPARM parameters for a CMAS, and SEC(YES) is coded for a MAS that is connecting to that CMAS, the attempt to establish the connection between the CMAS and the MAS fails. The following message appears in the EYULOG of the CMAS:

```
EYUCR0007E 'Security mismatch between CMAS EYUCMS1A and MAS EYUMAS1A .
          Connection Terminating.'
```

It is also possible to terminate the connection between a CMAS and a MAS using the STOP action command on the MAS view.

The preceding causes of the INACTIVE status have not dealt with the case where a CICSplex is managed by multiple CMASs. Consider the CICSplex shown in Figure 52.

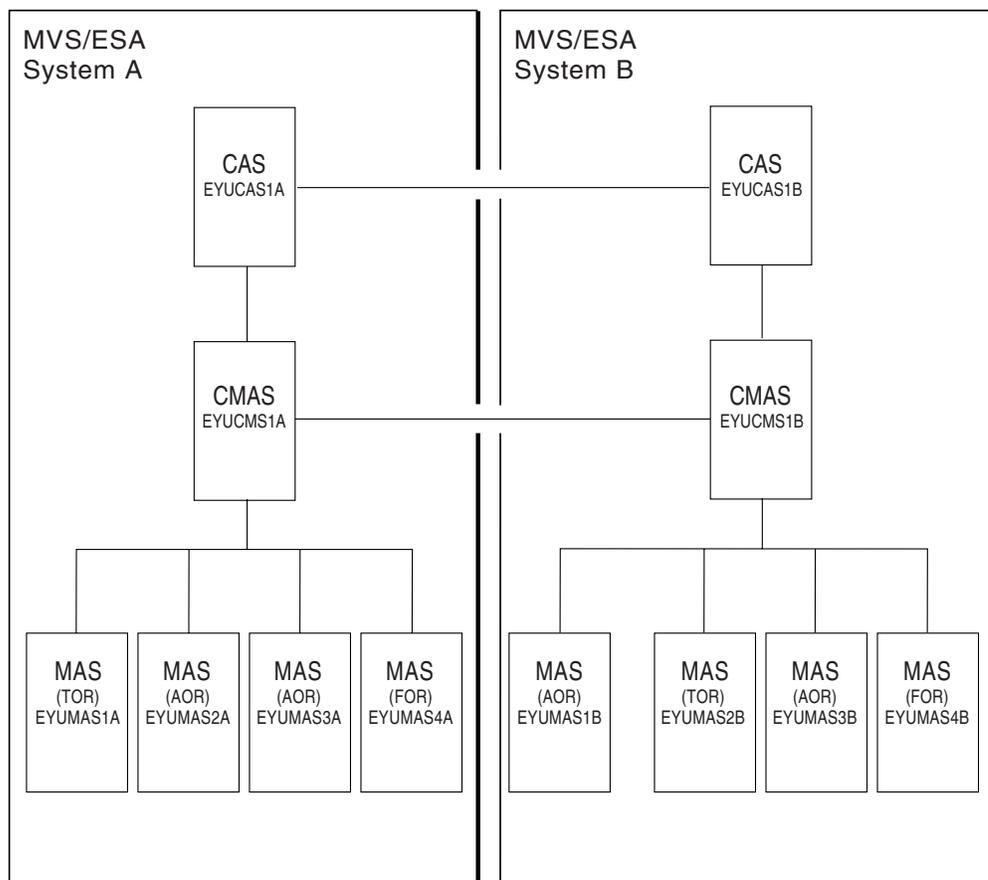


Figure 52. Example of a CICSplex managed by multiple CMASs

Let's say you are connected to CICSplex SM with a context of EYUPLX01 and your server CMAS is EYUCMS1A. You could use the following command to specify that context:

```
CONtext EYUPLX01 CPSM EYUCMS1A
```

You know that all eight MAS regions are running, yet a MAS command with a scope of EYUPLX01 returns the following:

```
27FEB2005 06:57:38 ----- INFORMATION DISPLAY -----
COMMAND ==>
CURR WIN ==> 1 ALT WIN ==>
>W1 =MAS=====EYUPLX01=EYUPLX01=27FEB2005==06:57:37=CPSM====8=====
CMD Name Type CMAS Status MON RTA WLM Description
-----
EYUMAS1A EYUCMS1A ACTIVE NO YES YES
EYUMAS2A EYUCMS1A ACTIVE NO YES YES
EYUMAS3A EYUCMS1A ACTIVE YES NO YES
EYUMAS4A EYUCMS1A ACTIVE YES NO YES
EYUMAS1B N/A INACTIVE N/A N/A N/A
EYUMAS2B N/A INACTIVE N/A N/A N/A
EYUMAS3B N/A INACTIVE N/A N/A N/A
EYUMAS4B N/A INACTIVE N/A N/A N/A
```

Figure 53. Example of failed connectivity between CMASs

## incomplete operations data returned

The result shown in Figure 53 on page 111 occurs if EYUCMS1B is not running, or if communication between EYUCMS1A and EYUCMS1B has failed. In general, the CMAS serving an end-user interface request must have connectivity to the CMAS to which a MAS is connected; if it does not, that MAS does not appear active to the end-user interface.

The CICSPLEX view shows (from the perspective of one CMAS) the connectivity to the other CMASs managing a CICSplex. To use this view, set the context to the CMAS that is serving your end-user interface session (EYUCMS1A), then issue the CICSPLEX command, as follows:

```
CONtext EYUCMS1A;CICSPLEX EYUPLX01
```

Figure 54 shows the result:

```
27FEB2005 07:41:08 ----- INFORMATION DISPLAY -----
COMMAND ==>>                                SCROLL ==>>
CURR WIN ==>> 1          ALT WIN ==>>
W1 =CICSPLEX=====EYUCMS1A=EYUPLX01=27FEB2005==07:40:42=CPSM====2=====
CMD CICSPlex CMAS      Status  MP  Access  Transit  Transit
-----
EYUPLX01 EYUCMS1A ACTIVE  YES  LOCAL          0
EYUPLX01 EYUCMS1B INACTIVE NO  N/A            0
```

Figure 54. Example of connectivity shown on the CICSPLEX view

If the CICSPLEX view shows a CMAS with INACTIVE status, but you know that CMAS is running, you must investigate the communication links. CMAS-to-CMAS communication uses CICS services. Therefore, the MSGUSR log is likely to contain information concerning the nature of the communication failure.

## ACTIVE status

An ACTIVE status indicates that a MAS is properly connected to the CICSplex. There should be no problem with missing data.

---

## Missing monitor data

This section discusses reasons why you might not receive monitor data from one of the MONITOR views. Here is an example of that:

```
27FEB2005 15:38:19 ----- INFORMATION DISPLAY -----
COMMAND ==>>                                SCROLL ==>>
CURR WIN ==>> 1          ALT WIN ==>>
W1 =MNTRATDQ=====EYUPLX01=EYUMAS1A=27FEB2005==15:38:19=CPSM=====
BBMXBD15I There is no data that satisfies your request
```

Figure 55. Example of the MNTRATDQ view with no monitoring information

The first step is to set the scope to the CICS system from which you are receiving no monitor data, and then issue the MONACTV view. The MONACTV view shows all active monitor definitions.

```

27FEB2005 15:48:22 ----- INFORMATION DISPLAY -----
COMMAND ==>>
CURR WIN ==>> 1          ALT WIN ==>>
WI =MONACTV=====EYUPLX01=EYUMAS1A=27FEB2005==15:48:22=CPSM====7=====
CMD Def      CICS      Status   Active   Resource Resource Include Res
--- Name---- System-- ----- Period-- Name---- Type--- ----- Stat
*            EYUMAS1A ACTIVE          *          MCONN   YES    NO
*0000004    EYUMAS1A ACTIVE          *          MFILE   YES    NO
*0000008    EYUMAS1A ACTIVE          CEMT     MTRAN   YES    NO
*0000010    EYUMAS1A ACTIVE          *          MPROG   YES    NO
HTTRAN      EYUMAS1A ACTIVE          *          MTRAN   YES    NO
ZDZMON2     EYUMAS1A ACTIVE          S123     MTERM   NO     NO
ZDZTERM     EYUMAS1A ACTIVE          S*       MTERM   YES    NO

```

Figure 56. Example of the MONACTV view showing active monitor definitions

Things to look for:

- Verify that the monitor definition has an active status. It is possible that a period definition is causing the monitor definition to be in a pending status.
- If there are multiple monitor definitions for a resource type, there may be a monitor definition that excludes the resource you expect to be monitored. So that all resource types are listed together, you may find it helpful to use the SORT display command on the Resource Type column. By grouping the resource types, you can see more easily whether a value of NO in the Include column is affecting the resource in question.
- Monitoring data is not accessible via the MONITOR views until one complete Sample Interval is complete. Therefore, depending on when a monitor definition was installed in relation to the sample interval cycle, you may have to wait through two sample intervals before monitoring data is accessible via the MONITOR views. Browse the MAS view to see what the sample interval is for each resource type.
- Verify that the monitor definition controlling the resource in question is in the list. If it is not, browse the MAS view to confirm that monitoring is active and that there is a nonzero sample interval for that particular resource type. Consider the example of the monitor section of a MAS view shown in Figure 57.

```

----- Browse Monitor for EYUPLX01 -----
COMMAND ==>>

MAS      EYUMAS1A  MON Active   YES

Retention period      0

Resource Sample Intervals:
CICS Region          120      Connections      900
Global               120      Files            900
DB2                  900      Journals         900
                               Programs         900
                               Terminals        30
                               Transactions     900
                               TD Queues        000

Type DOWN or UP to view other MAS screens.
Type END or CANCEL to terminate browse

```

Figure 57. Example of the monitor section of the MAS view

## missing monitor data

The view shown in Figure 57 on page 113 shows that monitoring is active for all resource types except transient data queues.

---

## Unexpected real-time analysis results

This section uses two sample problems to discuss ways to approach unexpected real-time analysis results. One problem deals with system availability monitoring (SAM), the other with MAS resource monitoring (MRM).

### An example SAM problem

A CICS system is known to be running and short on storage, yet the condition does not show up in the EVENT view.

1. Verify that the MAS view shows an active status for that CICS system.

```
27FEB2005 02:57:49 ----- INFORMATION DISPLAY -----
COMMAND ===>                                SCROLL ===>
CURR WIN ===> 1          ALT WIN ===>
W1 =MAS=====EYUPLX01=EYUMAS1A=27FEB2005==02:57:49=CPSM===1=====
CMD Name      Type  CMAS      Status  MON RTA WLM Description
---          -
EYUMAS1A LOCAL  EYUCMS1A ACTIVE  YES YES YES TOR1 - SYSA
```

Figure 58. MAS view showing active CICS system

If the MAS view does not show an active status, see “Incomplete operations data returned” on page 109.

2. Verify that the RTA Active field on the MAS view indicates YES. This is required for CICSplex SM to perform system availability monitoring for any of the predefined conditions (SOS, SYSDUMP, TRANDUMP, MAXTASK, STALL). You can make real-time analysis active immediately by using the UPD line command, and setting the RTA Active field on the first panel to YES. To make the change permanent, you must update the CICSSYS definition. For more information about the CICSSYS view command, see *CICSplex System Manager Administration*.
3. Determine which action definition controls what happens for the short-on-storage (SOS) condition. To do this, use the BRO line command to browse the MAS information. Scroll down to the Browse Analysis panel.

```

----- Browse Analysis for EYUPLX01 -----
COMMAND ==>

MAS      EYUMAS1A  RTA Active      YES
Primary CMAS      EYUCMS1A  CMAS Name
Active Period      PRIME      Period Def

System availability management:

          Action      Severity
SAM
SOS      SOSPRI1      VHS
SYSDUMP
TRANDUMP
MAXTASK
STALL

Type DOWN or UP to view other MAS screens.
Type END or CANCEL to terminate browse
    
```

Figure 59. Example of the Browse Analysis panel for the MAS view

In the example shown in Figure 59, the SOS condition is controlled by an action definition called SOSPRI1, and the other conditions assume the default actions. The default action is to issue a CICSplex SM EVENT and to send condition entry and condition exit WTO messages.

4. Use the ACTNDEF command to see which type of external notification is supposed to be issued for an SOS condition.

```

27FEB2005 16:00:54 ----- INFORMATION DISPLAY -----
COMMAND ==>                                SCROLL ==>
CURR WIN ==> 1          ALT WIN ==>
W1 =ACTNDEF=====EYUPLX01=EYUPLX01=27FEB2005==16:00:54=CPSM====1=====
CMD Name      Event      View      Msg      Alert      ARM      Description
-----
SOSPRI1      NO              YES      YES      NO      Critical SOS on production
    
```

Figure 60. Example of the ACTNDEF view

In the example shown in Figure 60, no CICSplex SM event is to be viewable from the EVENT view.

## An example MRM problem

MAS resource monitoring (MRM) can be used to generate an event whenever any of a particular group of transactions is disabled in a particular MAS. The LOCTRAN view (with scope set to that MAS) shows that one of the transactions is disabled, yet no event shows up in the EVENT view.

1. Verify that the real-time analysis definition is active. Set the scope to the MAS in question and issue the RTAACTV view command.

## unexpected RTA results

```
27FEB2005 04:53:08 ----- INFORMATION DISPLAY -----
COMMAND ==>
CURR WIN ==> 1 ALT WIN ==>
WI =RTAACTV=====EYUPLX01=EYUMAS1A=27FEB2005==04:53:08=CPSM====6=====
CMD Name System Status Period Rate Action Def
-----
DSAGETMN EYUMAS1A PENDING TVSHIFT2 60 DSAGMACT RTADEF
TRANDIS EYUMAS1A ACTIVE 60 DSALWACT RTADEF
LFILEDEL EYUMAS1A PENDING TVSHIFT2 300 LFILDACT RTADEF
LFILEOPN EYUMAS1A ACTIVE 300 LFILOACT RTADEF
PGMUSE EYUMAS1A ACTIVE 60 PGMUSACT RTADEF
PGM1 EYUMAS1A PENDING TVSHIFT2 60 PGMUSACT RTADEF
```

Figure 61. Example of the RTAACTV view showing active analysis definitions

If the analysis definition is not in the list, or is in the list with a PENDING status, that explains why nothing shows up in the EVENT view. The PENDING status indicates that the analysis definition is not within the Period shown. Absence from this active list indicates the analysis definition was either discarded (by use of the DSC line command on the RTAACTV view) or never installed.

2. Examine the analysis definition and related evaluation definitions and action definitions. If the analysis definition is listed in the RTAACTV view, you should reexamine the analysis definition, the evaluation definitions that make up the analysis definition's evaluation expression, and the associated action definitions.

Here are some points to consider:

- a. Sample Interval

The sample interval affects how soon the occurrence of a particular condition (such as a transaction becoming disabled) results in a real-time analysis notification. Also keep in mind that there are two sample intervals: the *evaluation* definition has a sample interval, which determines how often a resource is sampled, and the *analysis* definition has a sample interval, which determines how often an evaluation expression is evaluated.

- b. Entry and Exit Intervals

An analysis definition's entry and exit intervals have an effect on when a real-time analysis notification follows the occurrence of a certain condition.

- c. Action definitions

You should ensure that the action definition associated with an analysis definition is set up to deliver the action that you expect. It is possible that a notification results in an SNA generic alert and not in an external message or a CICSplex SM event.

---

## Unexpected workload management routing decision

You may need to investigate questionable or misunderstood dynamic routing decisions. For example, you might expect a specific dynamic routing request to be routed to the healthiest target region in a group of target regions. However, you might find that the request is always routed to one particular target region, regardless of the health of the target region.

The approach described here is as follows:

1. Make sure that dynamic routing is enabled for the work requests
2. Determine which workload is active
3. Determine whether the workload is separated by TRANSID, LUNAME or USERID
4. Determine whether there are active affinities

## Is dynamic routing enabled?

You should check the following:

- In the transaction definition, the Dynamic and Routable fields should be set to Yes.
- In the program definition, the Dynamic field should be set to Yes.
- The program should not be defined to the local system.
- The program may not be picking up the correct transaction id. Transaction ids are selected in the following order of precedence:
  - The transaction id specified in the EXEC CICS LINK command takes priority over a transaction id supplied in any other way.
  - The transaction id supplied in EYU9WRAM, the communication area for the dynamic routing user exit EYU9XLOP.
  - The transaction id specified in the program definition, if there is no transaction id specified in either the EXEC CICS LINK command or EYU9WRAM.
  - By default, if all other possibilities are blank, the CICS mirror transaction CSMI.

## Which workload is active?

The first step is to determine which workload is active in the region from which the dynamic request is routed. Issue the WLMSCOPE command.

```

27FEB2005 08:40:03 ----- INFORMATION DISPLAY -----
COMMAND ==>
CURR WIN ==> 1          ALT WIN ==>
W1 =WLMSCOPE=====EYUPLX01=EYUPLX01=27FEB2005==08:40:03=CPSM====9=====
CMD WLM      Scope      Scope      Scope      Scope      Update
--- Spec---- Name---- Type---- Mode---- Link---- Option--
EYUWLS01 EYUMAS1A CICSSYS INHERIT EYUCSG01
EYUWLS01 EYUMAS2A CICSSYS INHERIT EYUCSG01
EYUWLS01 EYUMAS3A CICSSYS INHERIT EYUCSG01
EYUWLS01 EYUCSG01 SYSGROUP
EYUWLS02 EYUMAS1B CICSSYS INHERIT EYUCSG02
EYUWLS02 EYUMAS2B CICSSYS INHERIT EYUCSG02
EYUWLS02 EYUMAS3B CICSSYS INHERIT EYUCSG02
EYUWLS02 EYUCSG02 SYSGROUP
EYUWLS02 EYUCSG03 SYSGROUP

```

Figure 62. Example of WLMSCOPE view showing active workloads

A routing region can be associated with only one workload specification. In the WLMSCOPE view, look in the Scope Name field for the routing region you are concerned with, and find the name of the associated workload specification. This name is the name of the workload that is activated when the requesting region starts.

One thing to remember about the WLMSCOPE view (and all other workload views) is that it reflects information that is in the data repository. It is possible that the data repository has been modified since its definitions were installed into running systems. Therefore, you must use the WLMAXxxx views to see which definitions are installed and active in running systems.

To verify that a workload is active, issue the WLMWORK view command.

## unexpected WLM routing decision

```
27FEB2005 08:59:19 ----- INFORMATION DISPLAY -----
COMMAND ==>>                                SCROLL ==>>
CURR WIN ==>> 1          ALT WIN ==>>
W1 =WLMWORK=====EYUPLX01=EYUPLX01=27FEB2005==08:59:19=CPSM====1=====
CMD Name      Ownr Rout Targ Affinity Lifetime Scope  Event  Status Algrthm
-----
----- Cnt-  Cnt-  -----
EYUWLS02 HTC1    1    2 LUNAME  LOGON  EYUMAS2B      ACTIVE QUEUE
```

Figure 63. Example of the WLMWORK view showing an active workload

Now you need to ensure that the workload is actively associated with the routing region you are interested in. To display the WLMAWTOR view, hyperlink from the Rout Cnt field on the WLMWORK view.

```
27FEB2005 09:03:20 ----- INFORMATION DISPLAY -----
COMMAND ==>>                                SCROLL ==>>
CURR WIN ==>> 1          ALT WIN ==>>
W1 =WLMAWTOR=====EYUPLX01=EYUPLX01=27FEB2005==09:03:20=CPSM====1=====
CMD Workload Ownr Router  Connection
-----
----- Lost-----
EYUWLS02 HTC1 EYUMAS2B
```

Figure 64. Example of the WLMAWTOR view

The WLMAWTOR view shows which routing regions are actively running a given workload.

## Is the workload being separated?

Now you know which workload is active on the routing region. The next step is to find out if the workload is being separated based on TRANSID, USERID, LUNAME, or some combination of these. To do that, take the request in question (the one defined as dynamic, initiated via terminal input) and see whether it is a member of any active transaction groups. Issue the WLMATRAN command.

```
27FEB2005 09:16:49 ----- INFORMATION DISPLAY -----
COMMAND ==>>                                SCROLL ==>>
CURR WIN ==>> 1          ALT WIN ==>>
W1 =WLMATRAN=====EYUPLX01=EYUPLX01=27FEB2005==09:16:49=CPSM====8=====
CMD Transid  PCONV  Trangrp  Workload Ownr
-----
----- Mode-  -----
ADCD          WMTAFFA  EYUWLS02 HTC1
DAA1          WMTAFFB  EYUWLS02 HTC1
DAA2          WMTAFFC  EYUWLS02 HTC1
DBA1          WMTAFFB  EYUWLS02 HTC1
DBA2          WMTAFFC  EYUWLS02 HTC1
DCA1          WMTAFFB  EYUWLS02 HTC1
DCA2          WMTAFFC  EYUWLS02 HTC1
F100          WMTMSCA  EYUWLS02 HTC1
```

Figure 65. Example of the WLMATRAN view

If the transaction in question is listed in this view, the routing decision is possibly based on a workload definition associated with the transaction group of which the transaction is a member. Note the name of the transaction group.

Now look at the active workload definitions. Issue the WLMWDEF command.

```

27FEB2005 11:42:55 ----- INFORMATION DISPLAY -----
COMMAND ==>
CURR WIN ==> 1          ALT WIN ==>
>W1 =WLMWDEF=====EYUPLX01=EYUPLX01=27FEB2005==11:42:55=CPSM====4=====
CMD Name      Workload Ownr  Trangrp  Luname      Userid  Target  Descrip
-----
T123DEF      EYUWLS02 HTC1      .++++T123   *        EYUMAS1B
WMDFAFFA     EYUWLS02 HTC1      WMTAFFA    .*        *        EYUMAS1B
WMDFAFFB     EYUWLS02 HTC1      WMTAFFB    .*        DEPT02*  EYUMAS2B
WMDFAFFC     EYUWLS02 HTC1      WMTAFFC    .*        *        EYUCSG02

```

Figure 66. Example of the WLMWDEF view

This view shows you which workload definition, if any, applies to the routing request in question. You know the USERID and LUNAME from which the routing request came. You also know whether the transaction is a member of an active transaction group, and, if it is, you know the name of the transaction group. Given these three things, you can tell which workload definition, if any, controls the routing decision. The following pseudo code explains the logic:

```

IF dynamic transaction in question is a member of an active transaction group
    THEN IF there is a workload definition associated with that transaction group
        THEN IF the USERID and NAME match the pattern on that workload definition
            THEN that workload definition will control the routing decision
            ELSE the workload default controls the routing decision
        ELSE the workload default controls the routing decision
    ELSE IF there is a workload definition not associated with a transaction group
        THEN IF the USERID and NAME match the pattern on that workload definition
            THEN that workload definition will control the routing decision
            ELSE the workload default controls the routing decision
        ELSE the workload default controls the routing decision

```

To illustrate this logic, here are some examples using the WLMWDEF view shown in Figure 66.

#### Example 1

The transaction is a member of active transaction group WMTAFFA. The USERID is DEPT01DZ. The LUNAME is NET1.IYJFT123. The routing decision is controlled by workload definition WMDFAFFA.

#### Example 2

The transaction is not a member of an active transaction group. The USERID is DEPT01DZ. The LUNAME is NET1.IYJFT123. The routing decision is controlled by workload definition T123DEF.

#### Example 3

The transaction is a member of active transaction group WMTAFFB. The USERID is DEPT01DZ. The LUNAME is NET1.IYJFT123. The routing decision is controlled by the workload default.

When you know which workload definition is controlling the routing decision, the Target Scope field on that same WLMWDEF view shows you the target region or target region group to which the transaction is routed. If the workload default is controlling the routing decision, the Target Scope field on the WLMWORK view shows where the transaction is routed.

### Are there any active affinities?

Given that a transaction is routed to a specific target region group, an active affinity forces the transaction to go to a particular target region in that group. Affinities are

## unexpected WLM routing decision

associated with a transaction group. To see whether there are any active affinities for a transaction group, issue the WLMATGRP command to show all active transaction groups. Then, hyperlink from the Affinity field. If there is no active transaction group involved, the default transaction group comes into play. To see whether there is an affinity associated with the default transaction group, hyperlink from the Affinity field of the WLMAWORK view.

---

## Application programming interface problems

If you are having problems with a program written using the CICSplex SM application programming interface (API), the first step is to rule out:

- Coding errors in the program itself.
- Incompatibilities between the program and the environment where it is running. Specifically:
  - If the API program is a REXX exec, ensure that the API function package (module EYU9AR00 with the aliases of EYU9AR01 and IRXFUSER) is in an authorized library that is in the MVS linklist or allocated to the STEPLIB DD in the address space in which the REXX exec is running.
  - If the API program is an assembled or compiled program, verify that the program assembled or compiled properly and that it was link-edited with the proper API stub for the environment in which the program will run. The API stub for a CICS environment is EYU9AMSI. The API stub for a non-CICS environment is EYU9ABSI.

If you have ruled out these potential sources of problems and the program still does not run successfully, you should do the following:

1. Check for error messages and abends.

Such messages could be issued by:

- The CMAS to which the API processing thread is connected.
- The MAS or user address space where the program is running.

If the program is running under MVS as a batch, TSO, or NetView® program, error messages are written to the MVS console. If the program is running under CICS, error messages are written to the CICS message log.

2. Collect the following documentation:

- Program source
- Program listing (for compiled or assembled programs)
- Linkage editor map (for compiled or assembled programs)

In addition, collect as much of the following as possible:

- AUXTRACE data set for the CMAS, if available
- Formatted EYU\_TRACE output (for REXX programs). Refer to the *CICSplex System Manager Application Programming Guide* book for details on EYU\_TRACE.
- System console log
- Appropriate job logs
- System or transaction dump, if any

When you have all the relevant information, contact your IBM Support Center.

---

## Chapter 14. Investigating Web User Interface problems

This topic contains the following sections:

- “Server and browser messages”
- “COVC status panel”
- “COVC debugging commands” on page 122
- “Typical end-user problems” on page 123

---

### Server and browser messages

During the operation of the Web User Interface, messages are written to the following destinations:

- EYULOG
- Console
- Browser

### Server messages

The Web User Interface server messages are mainly written to the CICSplex SM EYULOG of the Web User Interface server (and not of the CMAS). Some messages are also written to the console. The Web User Interface server messages are explained in the Web User Interface Message Help.

### Browser messages

The following types of Web User Interface messages are written to the browser:

- Client

These messages reflect status during the operation of the Web User Interface.

- Editor

These messages reflect status during the operation of the View Editor.

- HTTP

These messages reflect the HTTP response codes. HTTP messages can sometimes be hidden with substitute messages being issued by the browser. For example, you may receive Internet Explorer 'friendly HTTP messages' in place of Web User Interface HTTP messages.

You can obtain help for the client and editor messages by clicking the Help icon to the right of the message or by accessing the contents page of the Web User Interface Message Help. You can obtain help for HTTP messages by accessing the contents page of the Web User Interface Message Help.

---

### COVC status panel

The COVC status panel returns status information about the Web User Interface server as shown in Figure 67 on page 122.

```

COVC                CICSPlex SM Web User Interface Control          EYUVCTS

                                Status Details

    CMAS Sysid          : QSTX
    Server Sysid        : QSGW
    CICSPlex SM Release : 0210

    Secure Sockets      : No
    Port                : 05126
    Hostname             : mvsxx.company.com

    TCP/IP Service Name : EYUWUI
    TCP/IP Service Status : Open
    TCP/IP Address       : 127.10.10.12

Current Status : Ready
Applid         : IYCQSTGW
Time           : 20:40:36
Date           : 02/27/2001

PF 1 Help      3 Exit
                                12 Return

```

Figure 67. COVC status panel

## COVC debugging commands

The COVC transaction provides access to the Web User Interface run-time environment. It can be used to format and manipulate the internal data structures of the Web User Interface.

### Running the COVC transaction for debugging

To run the COVC transaction, log onto the Web User Interface server and enter the COVC transaction ID, together with one of the following debugging commands:

- **START**  
This starts the Web User Interface server if not already started during PLTPI processing.
- **STOp**  
This shuts down the Web User Interface server.
- **TRace**  
This displays the COVC trace flags panel, as shown in Figure 11 on page 37, giving you the opportunity to set appropriate trace levels.
- **Dump**  
This displays the Web User Interface control blocks using the CODB memory display (for information about the CODB transaction see the *CICSPlex SM Problem Determination*). You can display a control block directly by specifying the control block on the COVC DUMP command, for example 'COVC DUMP ANCHOR' displays the global anchor block. The control blocks that you can display are listed in Table 7.

Table 7. Dump control blocks

Dump control blocks	Meaning
---------------------	---------

Table 7. Dump control blocks (continued)

Anchor	Global anchor block
Gslrt	Global task block
Res	NLS resource block
Mos	Managed object block
Mos VOMO object	Managed object block for object object
Mos VOMAobject attribute	Managed object block for attribute attribute of object object
Mos VOMX object action	Managed object block for action action of object object
Mos VOMP object action parameter	Managed object block for parameter parameter for action action of object object
View	View cache block
Cwi	Web interface block
Tasks	User tasks block
EYU0Vccc	Entry point and module header for named method within EYU9VKEC load module.

**Attention:** The CICSplex SM COVC DUMP keyword should be used only at the request of your IBM support center. You must take steps to ensure that this transaction is used only by authorized personnel because of the access to system control areas that it provides. Improper or unauthorized use of COVC DUMP may have serious consequences, including without limitation loss of data or system outage. Customers are solely responsible for such misuse.

**Note:** In both the COVC commands and the COVC DUMP commands listed in Table 7 on page 122, the characters written in uppercase are the minimum number of characters you need to type to issue the command.

---

## Typical end-user problems

Table 8 describes some typical end-user problems that you may encounter and offers possible solutions.

Table 8. Typical end-user problems

Problems	Possible solutions
User unable to sign on, even when specifying reconnect	Check HTTP cookie support is enabled in the user's browser.
Apparently random characters displayed on browser	Ensure correct codepage translation table (DFHCNV) is in use. See <i>CICSplex System Manager Web User Interface Guide</i> for information.
Can't see graphical attribute presentations	Check that you are using a Java-enabled browser and that it is enabled.
Message EYUVC1200E displayed	Ensure that the user's browser has HTML frame support and that it is enabled.

Table 8. Typical end-user problems (continued)

Message EYUVH0400E displayed (or HTTP 400 message)	<p>Ensure that the HTTP request is set up correctly:</p> <ul style="list-style-type: none"> <li>• Check 'Unescape' option of the CICS Web Interface is set to 'No', if you are using a pre-CICS Transaction Server for OS/390 version 1.3 CICS system</li> <li>• Check that your browser is using HTTP 1.0 or 1.1</li> <li>• Check your browser service level is appropriate</li> </ul>
Message EYUVH0503E displayed (or HTTP 503 message) or no response from server	Check that your Web User Interface server is active and the CICS Web Interface operational.
Message EYUVH404E displayed (or HTTP 404 message)	Check that the URL being used to access the server was entered correctly and valid. If accessing a help page ensure that the help page exists.
Missing data fields in views	Some attribute fields are derived from CICS CMF performance class monitoring data.lin order for these fields to function correctly, you need to ensure that the CICS monitoring facility is active by setting the CICS system initialization parameters MNPER and MNRES to YES.

For further information about the messages access the contents page of the Web User Interface Message Help. For information about how to access the contents page of the Web User Interface Message Help, see *CICSplex System Manager Web User Interface Guide*. For the client message you can also obtain help by clicking the Help icon to the right of the message.

---

## **Part 4. Working with IBM to solve a problem**

This part of the book explains how to work with IBM Program Support to solve a CICSplex SM problem.



---

## Chapter 15. IBM Program Support

The IBM Customer Engineering Program Support structure exists to help you resolve problems with IBM products, and to ensure that you can make the best use of your IBM computing systems. Program support is available to all licensed users of IBM licensed programs. You can get help with your IBM program by calling your local Support Center.

This chapter will help you decide when to contact the Support Center, and what information you need to collect before contacting the Center. The chapter will also help you understand how IBM Program Support works.

---

### When to contact the Support Center

Before contacting the Support Center, try to ensure that the problem cannot be resolved without IBM's attention. In practice, many of the problems reported to Program Support turn out to be user errors. Other reported problems either cannot be reproduced or need to be handled by other parts of IBM Service, such as Hardware Customer Engineering or Systems Engineering. This indicates just how difficult it can be to determine the exact cause of a problem.

If you have followed the suggestions in this book and investigated all possible causes without finding the answer to your problem, then it is time to contact the Support Center.

---

### Working with the Support Center

When you call the Support Center, your first contact will be with a Support Center operator. This operator records some initial details about your problem, and then places it in a problem queue. You will receive a call back from a Support Center representative, who will try to help you solve your problem or refer you to someone who can.

The Support Center will need to know as much as possible about your problem. You should have the following information ready before making your first call:

- Your organization's name
- Your IBM Support Services access code
- The suspected source of the problem
- The severity level of the problem
- A complete description of the problem

Figure 68 on page 128 provides an overview of what happens when you call the Support Center.

## working with support

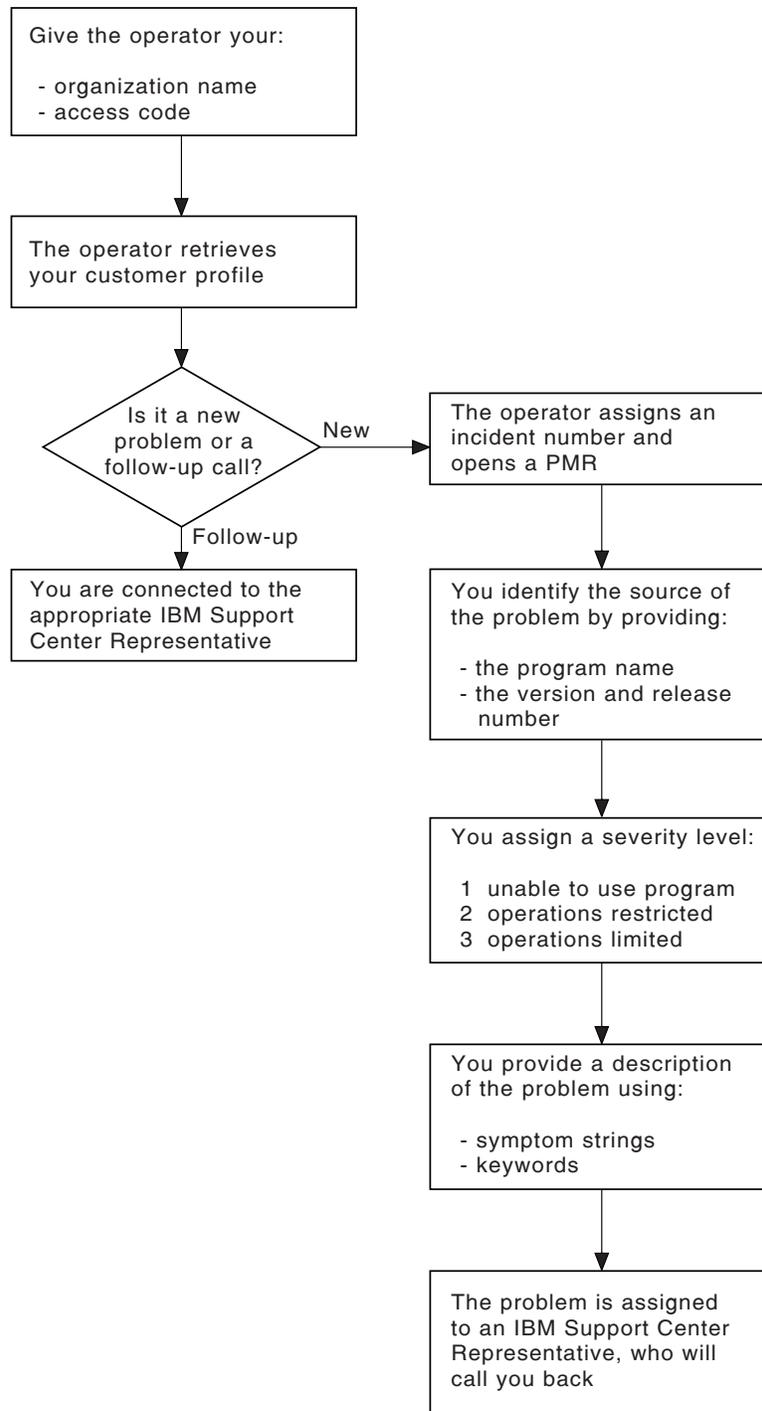


Figure 68. What happens when you call the Support Center

## What the Support Center needs to know

After you identify your organization, the operator will ask if you are calling about a new problem or following up on an existing problem. If it is a follow-up call, you will be connected with the appropriate Support Center representative.

If the problem is a new one, you will be assigned a unique incident number. A problem management record (PMR) will be opened on the RETAIN database, where all activity associated with your problem is recorded. Be sure to make a note of the incident number. You will be asked to provide the incident number in all future calls to the Center connected with this problem.

The operator will then ask you for the following information about the problem:

#### **The source of the problem**

The program that you suspect of causing the problem. Give the operator the program name, as well as its version and release number (for example, CICSplex SM Version 1 Release 2).

#### **Severity level**

A problem severity of 1, 2, or 3, based on the following conditions:

- Severity level 1 indicates that you are unable to use the program, resulting in a critical condition that needs immediate attention.
- Severity level 2 indicates that you are able to use the program, but that operation is severely restricted.
- Severity level 3 indicates that you are able to use the program, with limited functions, but the problem is not critical to your overall operation.

If the problem has a severity level of 1, it will normally be dealt with immediately.

#### **Description of the problem**

Use any CICSplex SM symptom strings or keywords associated with the problem. The primary keywords are WAIT, LOOP, PERFORMANCE, INCORROUT, and ABEND, corresponding to the problem classification types used earlier in this manual. Other keywords, that are not predefined, can also be useful, such as STARTUP, INITIALIZATION, a message or message number, an abend code, or any parameters known to be associated with the problem.

The keywords you supply will be used as search arguments on the RETAIN database. This helps the Support Center determine whether your problem is a known one that has already been the subject of an Authorized Program Analysis Report (APAR).

## **What happens next**

Details of your call are passed to the appropriate support group using the RETAIN problem management system. Your problem is put in the CICSplex SM problem queue. Problems are dealt with in order of severity level.

A Level 1 support representative will use the keywords you provided to search the RETAIN database. If your problem is found to be a known problem, and a fix has been developed for it, a Program Temporary Fix (PTF) will be sent to you.

If the RETAIN search is unsuccessful, your problem will be passed on to a Level 2 representative, who will contact you for more information about the problem.

Be sure to tell the support representative if any of the following events occurred before the problem appeared:

- Changes in level of MVS, CICS, or other licensed programs
- Regenerations
- PTFs applied
- New or additional features used

## working with support

- Application programs changed
- Unusual operator action.

You might be asked to quote values from a formatted dump or trace table. You might also be asked to perform some special task, such as turn on a specific level of trace, and then report on the results.

It might be necessary to have several follow-up calls, depending on the complexity of the symptoms and your system environment. The Support Center representative will record in the PMR the actions taken by you and IBM at every stage of problem diagnosis. The representative can then be acquainted with the complete history of the problem before making any follow-up calls.

The results of the investigation will determine whether your problem is really a new one, or one that is already known to IBM. If it is already known, and a fix has been developed, the fix will be sent to you.

If the problem is new, an APAR might be submitted. The APAR will be handled by the CICSplex SM change team. What you have to do is described in Chapter 16, “APARs, fixes, and PTFs,” on page 131.

---

## Chapter 16. APARs, fixes, and PTFs

An Authorized Program Analysis Report (APAR) is your means of informing the appropriate change team of a problem you have found in the current, unaltered release of an IBM program. When the change team solves the problem, they will produce a fix that will let you get your system running properly again. Finally, a Program Temporary Fix (PTF) is produced to replace the module in error, and the APAR is closed.

This chapter describes the APAR process, explains how to provide documentation in support of an APAR, and explains how to apply a PTF to CICSplex SM.

---

### The APAR process

The first step in the APAR process is for a Level 2 representative to enter the APAR into the RETAIN system. The APAR text consists of a detailed description of the problem. Your name is also entered with the APAR, so that the Support Center knows who to contact if the change team needs any further information.

When the APAR has been entered, you will be given an APAR number. You must write this number on all documentation you submit to the change team. This number will always be associated with the APAR and its resolution and, if a code change is required, with the fix as well.

The next step of the APAR process is up to you. You must submit to the change team all the documentation related to the APAR. Then, when the change team develops a PTF for the problem, you should apply the fix and test it on your system.

Here is a summary of the things you need to do:

1. Collect all the documentation that is required for the APAR. The Level 2 representative you are working with will tell you what you need. The documentation that is required varies, depending on the problem area, but “Collecting the documentation for an APAR,” will give you an idea of the material you may need to supply.
2. Package all the APAR documentation and send it to the change team. The procedure for this is described in “Sending the documentation to IBM” on page 132.
3. Apply the PTF that results from the APAR, preferably after testing the fix on your system. This step is described in “Applying the fix” on page 133.

---

### Collecting the documentation for an APAR

As a general rule, the documentation you submit for an APAR should include all the material you would normally use to do problem determination. Some of the documentation is common to all CICSplex SM problems, and some is specific to particular types of problems.

Make sure the problem you have described can be seen in the documentation you send. If the problem has ambiguous symptoms, you should document the sequence of events leading up to the failure. Tracing is valuable in this respect, but you might be able to provide details that a trace cannot give. You are encouraged to annotate your documentation, provided your annotation is legible and does not cover up any vital information. You can highlight data in any hard copy you send, using

## collecting documentation

transparent highlighting markers. You can also write notes in the margins, preferably using a red pen so that the notes are not overlooked.

Remember, if you send too little documentation, or if it is unreadable, the change team will have to return the APAR. Be sure to prepare your documentation carefully and send everything related to the problem.

Guidelines for the general documentation you will need are shown in “Documentation needed for problems with CICSplex SM.” However, you must find out from the Level 2 representative exactly what documentation you need to send for your specific problem.

## Documentation needed for problems with CICSplex SM

Here is a list of the general documentation you might be asked to submit for a CICSplex SM APAR:

- Any hard or soft copy illustrating the symptoms of the problem.
- A system dump of the CICS address space. Format the whole system dump if you plan to submit hard copy. Otherwise, you can send the system dump data set on tape.
- Output from the trace facilities.
- Console and other system logs.
- Listings of any relevant application programs and details of any user modifications.

For example: if you are using the EXEC CPSM application programming interface (API); if you are calling or customizing the dynamic transaction routing program, EYU9WRAM; or if you are using real-time analysis status probes, the Support Center needs a source listing of the relevant code that matches the executable version.

- JCL listings.
- A list of PTFs and APARs applied.

---

## Sending the documentation to IBM

The APAR documentation you submit for the problem can be sent electronically to an address provided by your IBM customer support representative or shipped in an APAR box, which you can obtain from your local IBM branch. APAR boxes are clearly marked as such, and they have a panel where tracking information, such as the APAR number, can be written.

Place all your documentation and notes in one or more APAR boxes. Be sure to mark all of the boxes, for example, ‘1 of 2’, and so on, if you need to use more than one.

If you include any magnetic tapes, indicate that clearly on the outside of the box. This will reduce the chances of the box being inadvertently stored in magnetic fields strong enough to damage the data.

Once you have packaged your documentation, a customer support representative will tell you where to send the package.

When the change team receives your package, it will be noted in your APAR record on the RETAIN system. The team will then investigate the problem. Occasionally, they will have to ask you for more documentation, perhaps specifying some trap you must set before getting it.

When the problem is solved, the APAR will be closed on the RETAIN system, and you will receive a fix.

At any time, you can contact the Support Center to find out how your APAR is progressing, particularly if it is a problem of high severity.

---

## Applying the fix

When the change team finds a fix for your problem, they might want you to apply it and test it on your system. For details on how to apply service to your system, refer to *CICS Transaction Server for z/OS Installation Guide*.

If they do ask you to test the fix, you are normally given two weeks to do it and provide them with feedback. However, you can ask for an extension if you are not able to complete the testing in that time.

When the change team is convinced that the fix is satisfactory, the APAR is certified by the CICSplex SM development team and the APAR is closed. You will be notified when this happens.

## The APAR becomes a PTF

If the problem solution involves a change to code in a CICSplex SM module that you can assemble, you will be sent the code change right away. The change will later be distributed as a PTF.

If you cannot assemble it yourself, because it involves a part of CICSplex SM that is distributed in object code only, you might be supplied with a ZAP or a TOTEST PTF.

If you need a PTF to resolve a specific problem, you can order it by its PTF number through the IBM Support Center. Otherwise, you can wait for the PTF to be sent out on the standard distribution tape. Instructions for applying service from the standard distribution tape are provided in *CICS Transaction Server for z/OS Installation Guide*.



---

## Part 5. Appendixes



---

## Appendix A. CICSplex SM naming standards

This appendix describes the naming standards used by CICSplex SM.

---

### The format of names

The names of modules, macros, and other source members distributed with CICSplex SM take the form:

prdtccxx

where:

- prd** Is a product code of BBC, BBM, or EYU.
- t** Identifies the type of element, as listed in “Element type identifiers.”
- cc** Is a component identifier, as listed in “Component identifiers” on page 138.
- xx** Is a unique identifier assigned by each component.

For example, EYUOMMIN is an executable module for the Monitor Services component.

### Element type identifiers

<b>ID</b>	<b>Description</b>
<b>\$</b>	Selection menus
<b>0</b>	Executable modules (C or assembler)
<b>5</b>	EUI record maps
<b>6</b>	Dynamically acquired control blocks or data areas
<b>7</b>	Module entry point descriptors
<b>8</b>	Function/service definition tables and assembled control blocks
<b>9</b>	Load modules
<b>B or R</b>	Assembler mapping DSECTs
<b>C</b>	C code generation macros
<b>D</b>	ISPF display or data entry panels
<b>E</b>	CLISTs
<b>F</b>	Function variables
<b>G</b>	ISPF message definitions
<b>H</b>	ISPF help panels
<b>J</b>	Screen definitions
<b>M</b>	C structure TYPEDEFs
<b>P</b>	Profile variables or USERFILE members
<b>Q</b>	Assembler code generation macros
<b>S</b>	EUI class tables
<b>T</b>	View, message, and action tables

## element type identifiers

<b>U</b>	Assembler equate files
<b>V</b>	C equate files
<b>W or X</b>	Assembled help modules
<b>Z</b>	View definitions

## Component identifiers

CICSplex SM component identifiers begin with one of three prefixes: BBC, BBM, or EYU.

### **BBC components**

The BBC components are:

<b>ID</b>	<b>Description</b>
<b>Qx</b>	PlexManager Data Collectors
<b>Sx</b>	Communications Server Controller
<b>Ux</b>	End-user Interface Address Space Services
<b>Zx</b>	Global Services

### **BBM components**

The BBM components are:

<b>ID</b>	<b>Description</b>
<b>Cx</b>	Data Manager
<b>Hx</b>	Information Services
<b>Lx</b>	Linkage Services
<b>Mx</b>	File Management
<b>Px</b>	Low-level Storage Management
<b>Qx</b>	PlexManager Selectors
<b>Sx</b>	General Services
<b>Tx</b>	TSO Support Functions
<b>Xx</b>	Transaction Management
<b>Zx</b>	System and Application Control

### **EYU components**

The EYU components are:

<b>ID</b>	<b>Description</b>
<b>Bx</b>	Business Application Services
<b>Cx</b>	Communications
<b>Ex</b>	End-user Interface
<b>Mx</b>	Monitor Services
<b>Nx</b>	Managed Application System
<b>Px</b>	real-time analysis
<b>Tx</b>	Topology Services

<b>Wx</b>	Workload Manager
<b>XC</b>	Data Cache Manager
<b>XD</b>	Data Repository
<b>XE</b>	Environment Services System Services
<b>XL</b>	Kernel Linkage
<b>XM</b>	Message Services
<b>XQ</b>	Queue Manager
<b>XS</b>	Common Services
<b>XZ</b>	Trace Services

---

## Major components of CICSplex SM

The major components of CICSplex SM and their 3-character identifiers are:

<b>Component Name</b>	<b>Identifier</b>
<b>Business Application Services</b>	BAS
<b>Common Services</b>	SRV
<b>Communications</b>	COM
<b>Data Cache Manager</b>	CHE
<b>Data Repository</b>	DAT
<b>Environment Services System Services</b>	ESS
<b>Kernel Linkage</b>	KNL
<b>Managed Application System</b>	MAS
<b>Message Services</b>	MSG
<b>Monitor Services</b>	MON
<b>Queue Manager</b>	QUE
<b>real-time analysis</b>	RTA
<b>Topology Services</b>	TOP
<b>Trace Services</b>	TRC

## major components

**Workload Manager**  
WLM

---

## Appendix B. System parameters for problem determination

CICSplex SM system parameters are used to identify or alter the attributes of a CMAS or MAS. Some system parameters are required in a CMAS or MAS startup job. However, the system parameters described here are optional and are used primarily for problem determination. In the course of diagnosing a problem, IBM customer support personnel may ask you to start up a CMAS or MAS with one or more of these parameters specified.

---

### Specifying system parameters

System parameters are specified by means of an extrapartition transient data queue with a destination ID of COPR. The parameters may be assigned to a DD \* file, sequential data set, or a partitioned data set member. The DD name for the extrapartition transient data queue is EYUPARM.

The parameters are coded as 80-byte records. Multiple system parameters can be specified on a single record as long as they are separated by commas and do not exceed 71 characters in length. The format of a system parameter is:

`keyword(value)`

where:

*keyword*

Is the name of a CICSplex SM system parameter.

There are two types of problem determination parameters for each CMAS or MAS component. The parameters are named as follows, where xxx is the 3-character component identifier:

#### **xxxTRACE**

Turns one or more levels of tracing on for the component. By default, component tracing is not active when a CMAS or MAS starts.

#### **xxxCONMSG**

Routes one or more levels of messages issued by the component to the console.

*value*

Is the alphanumeric data value assigned to the parameter.

For the trace and message parameters shown here, you can specify one or more values between 1 and 32. Values of 1 and 2 provide standard trace entries and messages; values of 3 through 32 cause special trace entries and messages to be recorded.

You can specify multiple values on a single parameter. To specify individual values, separate the values with a comma. To specify a range of values, separate the low and high values with a colon. For example:

```
KNLTRACE(1:3,16,28:32)
```

turns on trace levels 1 through 3, 16, and 28 through 32 in the Kernel Linkage (KNL) component.

To request multiple values for the same parameter, you must specify them as a single entry. If the same parameter is specified more than once, only the last entry is used.

## specifying system parameters

**Note:** Once a CMAS or MAS has been started, you can control the trace settings in a component by using the:

- CMAS or CMASD view to change CMAS component trace settings
- MAS view to change MAS component trace settings

For a description of these views, see *CICSplex System Manager Operations Views Reference*.

---

## The problem determination parameters

Table 9 lists the CICSplex SM system parameters that you may be asked to use for problem determination. As indicated in the table, some of the parameters can be used in the startup job for both CMASs and MASs; other parameters are specific to either a CMAS or a MAS.

Table 9. System parameters for problem determination

Name	Description	Values	Used by
BASTRACE	Business Application Services trace settings	1–32	Both
BASCONMSG	Business Application Services console message flags	1–32	CMAS
CHETRACE	Data Cache Manager trace settings	1–32	Both
CHECONMSG	Data Cache Manager console message flags	1–32	Both
CICSDUMPS	CICS system dumping active	NO   YES	CMAS
COMTRACE	Communications trace settings	1–32	Both
COMCONMSG	Communications console message flags	1–32	Both
DATTRACE	Data Repository trace settings	1–32	Both
DATCONMSG	Data Repository console message flags	1–32	Both
ESDUMP	Controls SVC dump suppression.	YES   NO   NEVER	Both <sup>3</sup>
ESDUMPCOM	Suppress communication task dumps	YES   NO	CMAS
ESDUMPLIMIT	Controls the number of dumps collected for a given failure.	0  999	CMAS
KNLTRACE	Kernel Linkage trace settings	1–32	Both
KNLCONMSG	Kernel Linkage console message flags	1–32	Both
MASTRACE	Managed Application System trace settings	1–32	MAS
MASCONMSG	Managed Application System console message flags	1–32	MAS
MONTRACE	Monitor Services trace settings	1–32	CMAS
MONCONMSG	Monitor Services console message flags	1–32	CMAS
MSGTRACE	Message Services trace settings	1–32	Both
MSGCONMSG	Message Services console message flags	1–32	Both
QUETRACE	Queue Manager trace settings	1–32	Both
QUECONMSG	Queue Manager console message flags	1–32	Both
RTATRACE	real-time analysis trace settings	1–32	Both <sup>1</sup>
RTACONMSG	real-time analysis console message flags	1–32	Both
SRVTRACE	Common Services trace settings	1–32	Both
SRVCONMSG	Common Services console message flags	1–32	Both
TOPTRACE	Topology Services trace settings	1–32	Both
TOPCONMSG	Topology Services console message flags	1–32	Both
TRCTRACE	Trace Services trace settings	1–32	Both
TRCCONMSG	Trace Services console message flags	1–32	Both
WLMTRACE	Workload Manager trace settings	1–32	Both <sup>2</sup>
WLMCONMSG	Workload Manager console message flags	1–32	Both

#  
#



## problem determination parameters

---

## Bibliography

---

### The CICS Transaction Server for z/OS library

The published information for CICS Transaction Server for z/OS is delivered in the following forms:

#### The CICS Transaction Server for z/OS Information Center

The CICS Transaction Server for z/OS Information Center is the primary source of user information for CICS Transaction Server. The Information Center contains:

- Information for CICS Transaction Server in HTML format.
- Licensed and unlicensed CICS Transaction Server books provided as Adobe Portable Document Format (PDF) files. You can use these files to print hardcopy of the books. For more information, see “PDF-only books.”
- Information for related products in HTML format and PDF files.

One copy of the CICS Information Center, on a CD-ROM, is provided automatically with the product. Further copies can be ordered, at no additional charge, by specifying the Information Center feature number, 7014.

Licensed documentation is available only to licensees of the product. A version of the Information Center that contains only unlicensed information is available through the publications ordering system, order number SK3T-6945.

#### Entitlement hardcopy books

The following essential publications, in hardcopy form, are provided automatically with the product. For more information, see “The entitlement set.”

### The entitlement set

The entitlement set comprises the following hardcopy books, which are provided automatically when you order CICS Transaction Server for z/OS, Version 3 Release 1:

*Memo to Licensees*, GI10-2559  
*CICS Transaction Server for z/OS Program Directory*, GI10-2586  
*CICS Transaction Server for z/OS Release Guide*, GC34-6421  
*CICS Transaction Server for z/OS Installation Guide*, GC34-6426  
*CICS Transaction Server for z/OS Licensed Program Specification*, GC34-6608

You can order further copies of the following books in the entitlement set, using the order number quoted above:

*CICS Transaction Server for z/OS Release Guide*  
*CICS Transaction Server for z/OS Installation Guide*  
*CICS Transaction Server for z/OS Licensed Program Specification*

### PDF-only books

The following books are available in the CICS Information Center as Adobe Portable Document Format (PDF) files:

#### CICS books for CICS Transaction Server for z/OS

##### General

*CICS Transaction Server for z/OS Program Directory*, GI10-2586  
*CICS Transaction Server for z/OS Release Guide*, GC34-6421  
*CICS Transaction Server for z/OS Migration from CICS TS Version 2.3*, GC34-6425

*CICS Transaction Server for z/OS Migration from CICS TS Version 1.3,*  
GC34-6423

*CICS Transaction Server for z/OS Migration from CICS TS Version 2.2,*  
GC34-6424

*CICS Transaction Server for z/OS Installation Guide,* GC34-6426

#### **Administration**

*CICS System Definition Guide,* SC34-6428

*CICS Customization Guide,* SC34-6429

*CICS Resource Definition Guide,* SC34-6430

*CICS Operations and Utilities Guide,* SC34-6431

*CICS Supplied Transactions,* SC34-6432

#### **Programming**

*CICS Application Programming Guide,* SC34-6433

*CICS Application Programming Reference,* SC34-6434

*CICS System Programming Reference,* SC34-6435

*CICS Front End Programming Interface User's Guide,* SC34-6436

*CICS C++ OO Class Libraries,* SC34-6437

*CICS Distributed Transaction Programming Guide,* SC34-6438

*CICS Business Transaction Services,* SC34-6439

*Java Applications in CICS,* SC34-6440

*JCICS Class Reference,* SC34-6001

#### **Diagnosis**

*CICS Problem Determination Guide,* SC34-6441

*CICS Messages and Codes,* GC34-6442

*CICS Diagnosis Reference,* GC34-6899

*CICS Data Areas,* GC34-6902

*CICS Trace Entries,* SC34-6443

*CICS Supplementary Data Areas,* GC34-6905

#### **Communication**

*CICS Intercommunication Guide,* SC34-6448

*CICS External Interfaces Guide,* SC34-6449

*CICS Internet Guide,* SC34-6450

#### **Special topics**

*CICS Recovery and Restart Guide,* SC34-6451

*CICS Performance Guide,* SC34-6452

*CICS IMS Database Control Guide,* SC34-6453

*CICS RACF Security Guide,* SC34-6454

*CICS Shared Data Tables Guide,* SC34-6455

*CICS DB2 Guide,* SC34-6457

*CICS Debugging Tools Interfaces Reference,* GC34-6908

### **CICSplex SM books for CICS Transaction Server for z/OS**

#### **General**

*CICSplex SM Concepts and Planning,* SC34-6459

*CICSplex SM User Interface Guide,* SC34-6460

*CICSplex SM Web User Interface Guide,* SC34-6461

#### **Administration and Management**

*CICSplex SM Administration,* SC34-6462

*CICSplex SM Operations Views Reference,* SC34-6463

*CICSplex SM Monitor Views Reference,* SC34-6464

*CICSplex SM Managing Workloads,* SC34-6465

*CICSplex SM Managing Resource Usage,* SC34-6466

*CICSplex SM Managing Business Applications,* SC34-6467

#### **Programming**

*CICSplex SM Application Programming Guide,* SC34-6468

*CICSplex SM Application Programming Reference,* SC34-6469

## Diagnosis

*CICSplex SM Resource Tables Reference*, SC34-6470  
*CICSplex SM Messages and Codes*, GC34-6471  
*CICSplex SM Problem Determination*, GC34-6472

## CICS family books

### Communication

*CICS Family: Interproduct Communication*, SC34-6473  
*CICS Family: Communicating from CICS on System/390*, SC34-6474

### Licensed publications

The following licensed publications are not included in the unlicensed version of the Information Center:

*CICS Diagnosis Reference*, GC34-6899  
*CICS Data Areas*, GC34-6902  
*CICS Supplementary Data Areas*, GC34-6905  
*CICS Debugging Tools Interfaces Reference*, GC34-6908

---

## Other CICS books

The following publications contain further information about CICS, but are not provided as part of CICS Transaction Server for z/OS, Version 3 Release 1.

<i>Designing and Programming CICS Applications</i>	SR23-9692
<i>CICS Application Migration Aid Guide</i>	SC33-0768
<i>CICS Family: API Structure</i>	SC33-1007
<i>CICS Family: Client/Server Programming</i>	SC33-1435
<i>CICS Transaction Gateway for z/OS Administration</i>	SC34-5528
<i>CICS Family: General Information</i>	GC33-0155
<i>CICS 4.1 Sample Applications Guide</i>	SC33-1173
<i>CICS/ESA 3.3 XRF Guide</i>	SC33-0661

---

## Determining if a publication is current

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

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

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

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



---

## Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully.

You can perform most tasks required to set up, run, and maintain your CICSplex SM system in one of these ways:

- using a 3270 emulator connected to CICSplex SM
- using a 3270 emulator logged on to CICS
- using a 3270 emulator logged on to TSO
- using a 3270 emulator as an MVS system console
- using the CICSplex SM web user interface.

IBM Personal Communications (Version 5.0.1 for Windows 95, Windows 98, Windows NT and Windows 2000; version 4.3 for OS/2) provides 3270 emulation with accessibility features for people with disabilities. You can use this product to provide the accessibility features you need in your CICSplex SM system.



---

# Index

## A

abend, investigating 105  
agent code, MAS 12  
ALLOC debugging command 63  
APAR (authorized program analysis report)  
  closing 133  
  documentation needed 131, 132  
  number 131  
  packaging 132  
  process 131  
  submitting 132, 133  
API program problems 120  
application programming interface problems 120  
applying a fix 133  
ATTACH debugging command 64  
authorized program analysis report (APAR)  
  closing 133  
  documentation needed 131, 132  
  number 131  
  packaging 132  
  process 131  
  submitting 132, 133  
AUXTR 36  
AUXTRACE facilities  
  in a CMAS 27  
  in a MAS 27

## B

bottleneck, investigating 107

## C

CALL debugging command 66  
CAPTURE debugging command 67  
CAS (coordinating address space) 5  
change log contents 21  
CICS AUXTRACE facilities  
  in a CMAS 27  
  in a MAS 27  
CICS system initialization parameters  
  AUXTR 36  
  SYSTR 36  
  USERTR 36  
CICS trace table settings 27  
CICSplex SM-requested dumps  
  during CMAS initialization 42  
  during ESSS PC routine execution 43  
  during MAS initialization 42  
  for a CAS 42  
CMAS  
  description 5  
  maintenance point 9  
  networks and registration 9, 10  
  structure 10, 11  
COD0 debugging commands  
  ALLOC 63

COD0 debugging commands (*continued*)

  ATTACH 64  
  CALL 66  
  CAPTURE 67  
  DUMP 68  
  EXEC 69  
  EXIT 70  
  HELP 70  
  LIST 70  
  POST 82  
  PRINT 82  
  PURGE 83  
  START 84  
  TRACE 84  
  TRACK 85  
  TRAP 86  
COD0 transaction 62  
CODB debugging commands 91, 92  
CODB transaction 90  
COLU transaction 57  
common components of CICSplex SM  
  Common Services 14  
  Communications 15  
  Data Cache Manager 14  
  Data Repository 14  
  Kernel Linkage 13  
  Message Services 13  
  Queue Manager 14  
  Trace Services 13  
common problems 18  
Common Services component 14  
Communications component 15  
communications tracing 34  
component identifiers 138  
  in element names 138  
  three-character 139  
controlling dumps 46, 47  
controlling the amount of trace  
  using system parameters 141  
  using the end-user interface 29, 30  
coordinating address space (CAS) 5  
COVC  
  debugging commands 122  
    DUMP 122  
    START 122  
    STOP 122  
    TRACE 122  
  dump 47  
  status details 121  
  trace flags 36

## D

Data Cache Manager component 14  
Data Repository component 14  
data spaces  
  description 12  
  dumping 44

- debugging transactions
  - method-level (COD0) 62
  - running 61
  - system-level (CODB) 90
- DIAGMSG view 97
- diagnostic documentation
  - for a bottleneck 107
  - for a stall 106
  - for an abend 105
  - for an APAR 131, 132
  - for problem determination
    - related products 21
    - site documentation 21
  - for the Support Center 129
- diagnostic facilities, PlexManager
  - DIAGMSG view 97
  - DIAGSESS view 98
  - DIAGSYS view 100
  - extended diagnostic mode (XDM) 101
- DIAGSESS view 98
- DIAGSYS view 100
- dump codes
  - EYU0VWAN 47
  - EYU0VWCV 47
  - WUIABEND 47
  - WUITRACE 47
- DUMP debugging command 68
- dump facilities
  - controlling 46, 47
  - IPCS tools
    - dump formatting routine 49, 53
    - types of dumps 39
- dump formatting routine 49, 53
- dump types
  - CICSplex SM-requested
    - during CMAS initialization 42
    - during ESSS PC routine execution 43
    - during MAS initialization 42
    - for a CAS 42
  - unexpected dumps
    - under CICS 40
    - under MVS 39
  - user-requested
    - using the end-user interface 45
    - using the MVS DUMP command 44
- dumps
  - Web User Interface 47

## E

- element types, CICSplex SM 137
- end-user interface 7, 9
- Environment Services System Services (ESSS)
  - description 6, 12
  - utility 53, 55
- ESSS (Environment Services System Services)
  - description 6, 12
  - utility 53, 55
- ESSS utility (EYU9XEUT)
  - JCL 54
  - options 53, 54

- exception trace 38
- exception tracing 29
- EXEC debugging command 69
- EXIT debugging command 70
- extended diagnostic mode (XDM)
  - disabling of error recovery 102
  - internal tracing 101
  - status messages 101
- EYU0XZPT system dump code 41
- EYU0XZSD system dump code 41
- EYU9XEUT utility
  - JCL 54
  - options 53, 54
- EYU9XZUT 36
- EYU9XZUT utility
  - JCL 33
  - options 31, 33
  - sample output 33

## F

- format of CICSplex SM names 137
- formatting dumps with IPCS 49, 53
- formatting options, trace 31, 33
- formatting trace entries 31, 34

## H

- help
  - Web User Interface message help 121
- HELP debugging command 70

## I

- IBM Support Center
  - follow-up calls 130
  - use of RETAIN system 129
  - what happens when you call 128
  - what they need to know 128, 129
  - when to contact 127
  - working with 127, 130
- incident number 129
- InfoManager component 5
- interpreting trace entries
  - for LU6.2 communications 35, 36
  - in a CMAS or MAS 30
- investigating specific problems
  - abends 105
  - bottlenecks 107
  - incomplete operations data 109, 112
  - missing monitor data 112, 114
  - stalls 106
  - unexpected RTA results 114, 116
  - unexpected WLM routing 116, 120
  - with the API 120
- IPCS dump formatting routine 49, 53
- IPCS VERBEXIT command 49
- ISPF end-user interface 7, 9

## K

Kernel Linkage component 13

## L

LIST debugging command

- ALLOC parameter 71
- CACHE parameter 72, 73
- CAPTURE parameter 74
- CLIST parameter 75
- COMM parameter 76
- METH parameter 77
- START parameter 78
- STCB parameter 79
- TASK parameter 79
- VIEWS parameter 81

logical records 22

LOGREC data set 22

LU6.2 tracing 34

## M

maintenance point CMAS 9

major object descriptor block (MODB) 11

major object environment block (MOEB) 11

MAL (message argument list) 11

managed application system (MAS)

- agent code 12
- description 5

MAS (managed application system)

- agent code 12
- description 5

MAS agent 12

message argument list (MAL) 11

Message Services component 13

messages 121

- as a source of information 22
- browser 121
- preliminary check for 18
- server 121

method call environment 11

method-level debugging with COD0

- allocating a resource 63
- attaching a method 64
- calling CICS programs 66
- capturing a view 67
- commands that alter CICSplex SM 62
- displaying a MAL 86
- displaying and altering data 68
- entering CODB 68
- executing a method 69
- exiting 70
- listing tasks and resources 70
- main menu 62
- online help 70
- PF key assignments 62
- posting an ECB 82
- printing data areas 82
- purging a resource 83
- recursive commands 62

method-level debugging with COD0 (*continued*)

- setting CICS and CICSplex SM trace flags 84
- setting trace flags based on call structure 85
- setting trace flags for a method 86
- starting a method in a CMAS 84

missing data fields 124

MODB (major object descriptor block) 11

MOEB (major object environment block) 11

monitor data, missing 112, 114

## N

naming convention 137, 141

## O

online diagnostic aids

- debugging transactions
  - method-level (COD0) 62
  - system-level (COdB) 90
- description 21

online utility (COLU) 57, 61

online utility, COLU 57

output problems 109, 112

overview of CICSplex SM

- agents in a MAS 12
- CMAS networks and registration 9, 10
- CMAS structure 10, 11
- common components 13, 15
- ESSS and data spaces 12
- ISPF end-user interface 7, 9
- structure of system 5, 6

## P

parameters, system

- for problem determination
  - list of 142
  - types 141
- specifying 141

PlexManager diagnostic facilities

- DIAGMSG view 97
- DIAGSESS view 98
- DIAGSYS view 100
- extended diagnostic mode (XDM) 101

PMR (problem management record) 129

POST debugging command 82

preliminary checks

- affecting specific parts of system 18
- changes since last run 17
- has system run before 17
- messages 18
- occurring at specific times 18

problem determination

- description 3
- preliminary checks 17, 18
- system parameters
  - list of 142
  - specifying 141
  - types 141

problem severity level 129

- problem types 18
- program support
  - follow-up calls 130
  - use of RETAIN system 129
  - what happens when you call 128
  - what they need to know 128, 129
  - when to contact 127
  - working with 127, 130
- program temporary fix (PTF)
  - applying 133
  - obtaining 133
- PTF (program temporary fix)
  - applying 133
  - obtaining 133
- PURGE debugging command 83

## Q

- Queue Manager component 14

## R

- RETAIN problem management system 129
- RTA results, unexpected
  - MAS Resource Monitoring 115, 116
  - System Availability Monitoring 114, 115

## S

- service point 9
- severity level, problem 129
- sources of information
  - change log 21
  - LOGREC records 22
  - manuals 21
  - messages 22
  - online diagnostic aids 21
  - site documentation 21
  - symptom strings 22
  - traces 23
- special trace levels 28
- stall, investigating 106
- standard trace levels 28
- START debugging command 84
- structure of a CMAS 10, 11
- structure of CICSplex SM 5, 6
- symptom strings 22
- symptoms of a problem 18
- SYS1.LOGREC data set 22
- SYSDUMP code entries in a MAS 41
- system management problems
  - missing monitor data 112, 114
  - unexpected RTA results 114, 116
  - unexpected WLM routing 116, 120
- system parameters
  - for problem determination
    - list of 142
    - types 141
  - specifying 141
- system-level debugging with CODB
  - accessing from COD0 95

- system-level debugging with CODB (*continued*)
  - altering memory 94
  - commands 91, 92
  - main menu 90
  - PF key assignments 92
- SYSTR 36

## T

- tools for problem determination
  - debugging transactions
    - method-level (COD0) 62
    - system-level (CODB) 90
  - dump facilities
    - controlling 46, 47
    - IPCS tools 49, 53
    - types of dumps 39
  - ESSS utility (EYU9XEUT) 53, 55
  - online utility (COLU) 57, 61
  - PlexManager facilities
    - DIAGMSG view 97
    - extended diagnostic mode (XDM) 101
- trace
  - Web User Interface 36
- TRACE debugging command 84
- trace facilities
  - controlling the amount of trace
    - using system parameters 141
    - using the end-user interface 29, 30
  - description of 23
  - for a CMAS 27
  - for a MAS 27
  - for LU6.2 communications 34
  - formatting trace entries 31, 34
  - interpreting trace entries
    - for LU6.2 communications 35, 36
    - in a CMAS or MAS 30
  - types and levels of trace 28, 29
    - exception 29
    - special 28
    - standard 28
  - use of CICS trace table 27
- trace flags 37
- trace format utility (EYU9XZUT)
  - JCL 33
  - options 31, 33
  - sample output 33
- trace formatting options 31, 33
- Trace Services component 13
- trace table, CICS 27
- tracing
  - exception 29
  - for LU6.2 communications 34
  - in a CMAS 27
  - in a MAS 27
  - special 28
  - standard 28
- TRACK debugging command 85
- TRANDUMP code entries in a MAS 41
- TRAP debugging command 86

- types of dumps
  - CICSplex SM-requested
    - during CMAS initialization 42
    - during ESSS PC routine execution 43
    - during MAS initialization 42
    - for a CAS 42
  - unexpected dumps
    - under CICS 40
    - under MVS 39
  - user-requested
    - using the end-user interface 45
    - using the MVS DUMP command 44
- types of problems 18

## U

- unexpected dumps
  - in a MAS 41
  - under CICS 40
  - under MVS 39
- user-requested dumps
  - using the end-user interface 45
  - using the MVS DUMP command 44
- USERTR 36
- using this book 3

## V

- VERBEXIT command 49

## W

- Web User Interface 6
  - dumps 41, 47
  - trace 36
  - typical problems 123
- Web User Interface message help 121
- Web User Interface server initialization parameters
  - WUITRACE 36
- WLM routing, unexpected 116, 120
- WUITRACE 36

## X

- XDM (extended diagnostic mode)
  - disabling of error recovery 102
  - internal tracing 101
  - status messages 101



---

## Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply in the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

---

## Programming interface information

The *CICSplex System Manager Problem Determination* book is intended to help you to do diagnosis of CICSplex SM. This book documents information that is Diagnosis, Modification, or Tuning information provided by CICSplex SM.

**Warning:** Do not use this Diagnosis, Modification, or Tuning information as a programming interface.

---

## Trademarks

IBM, the IBM logo, and [ibm.com](http://ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at Copyright and trademark information at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

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

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other product and service names might be trademarks of IBM or other companies.

---

## Sending your comments to IBM

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book.

Please limit your comments to the information in this book and the way in which the information is presented.

To ask questions, make comments about the functions of IBM products or systems, or to request additional publications, contact your IBM representative or your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

You can send your comments to IBM in any of the following ways:

- By mail, to this address:

IBM United Kingdom Limited  
User Technologies Department (MP095)  
Hursley Park  
Winchester  
Hampshire  
SO21 2JN  
United Kingdom

- By fax:
  - From outside the U.K., after your international access code use 44-1962-816151
  - From within the U.K., use 01962-816151
- Electronically, use the appropriate network ID:
  - IBMLink: HURSLEY(IDRCF)
  - Internet: idrcf@hursley.ibm.com

Whichever you use, ensure that you include:

- The publication title and order number
- The topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.







Product Number: 5655-M15

GC34-6472-05



Spine information:



CICS TS for z/OS

CICSplex SM Problem Determination

Version 3  
Release 1