CICS® Transaction Server for OS/390®

CICSPlex® SM Problem Determination

Release 3

GC33-0791-03

CICS® Transaction Server for OS/390®

CICSPlex® SM Problem Determination

Release 3

GC33-0791-03

Note!

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

Fifth Edition, March 1999

This edition applies to Release 3 of CICS Transaction Server for OS/390, program number 5655-147, and to any subsequent versions, releases, and modifications until otherwise indicated in new editions. Information in this edition was previously contained in GC33-0791-02, which is now obsolete. Make sure you are using the correct edition for the level of the product. The technical changes for this edition are summarized under “Summary of changes,” and are indicated by a vertical bar to the left of the change.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the addresses given below.

At the back of this publication is a page titled “Sending your comments to IBM”. If you want to make comments, but the methods described are not available to you, please address your comments to:

IBM United Kingdom Laboratories,
Information Development, Mail Point 095,
Hursley Park, Winchester, Hampshire,
England, SO21 2JN.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply in the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Programming interface information

The *CICSPlex SM 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

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

AT
CICS
CICS/ESA
CICSPlex
DB2
DXT
IBM
IMS
Language Environment
MVS/ESA
NetView
OS/2
OS/390
RETAIN
SP
VSE/ESA
VTAM

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

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

Preface

This book is intended to help you determine the cause of problems in a system running CICSPlex ® System Manager for CICS® Transaction Server for OS/390®. 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 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 OS/390. 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 may be used to control CICS systems that are directly connected to it, and indirectly connected through a previous release of CICSPlex SM.

For this release of CICSPlex SM, the directly-connectable CICS systems are:

- CICS Transaction Server for OS/390 1.3
- CICS Transaction Server for OS/390 1.2
- CICS Transaction Server for OS/390 1.1
- CICS for MVS/ESA 4.1
- CICS Transaction Server for VSE/ESA Release 1
- CICS for VSE/ESA 2.3
- CICS for OS/2 3.1
- CICS for OS/2 3.0

CICS systems that are not directly connectable to this release of CICSPlex SM are:

- CICS for MVS/ESA 3.3
- CICS for MVS 2.1.2
- CICS for VSE/ESA 2.2
- CICS/OS2 2.0.1

Note: IBM Service no longer supports these CICS release levels.

You can use this release of CICSPlex SM to control CICS systems 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 OS/390: Migration Guide* for information on how to do this.

Table 1 shows which CICS systems may 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 1.3	CICSPlex SM 1.3	CICSPlex SM 1.2
CICS TS 1.3	Yes	No	No
CICS TS 1.2	Yes	Yes	No
CICS TS 1.1	Yes	Yes	Yes
CICS for MVS/ESA 4.1	Yes	Yes	Yes
CICS for MVS/ESA 3.3	No	Yes	Yes
CICS for MVS 2.1.2	No	Yes	Yes
CICS TS for VSE/ESA Rel 1	Yes	No	No
CICS for VSE/ESA 2.3	Yes	Yes	Yes
CICS for VSE/ESA 2.2	No	Yes	Yes
CICS for OS/2 3.1	Yes	No	No
CICS for OS/2 3.0	Yes	Yes	Yes
CICS/OS2 2.0.1	No	Yes	Yes

Bibliography

CICS Transaction Server for OS/390

CICS Transaction Server for OS/390: Planning for Installation	GC33-1789
CICS Transaction Server for OS/390: Release Guide	GC34-5352
CICS Transaction Server for OS/390: Migration Guide	GC34-5353
CICS Transaction Server for OS/390: Program Directory	GC33-1706
CICS Transaction Server for OS/390: Licensed Program Specification	GC33-1707

CICS books for CICS Transaction Server for OS/390

General	
CICS Master Index	SC33-1704
CICS User's Handbook	SX33-6104
CICS Glossary (softcopy only)	GC33-1705
Administration	
CICS Transaction Server for OS/390: Installation Guide	GC33-1681
CICS System Definition Guide	SC33-1682
CICS Customization Guide	SC33-1683
CICS Resource Definition Guide	SC33-1684
CICS Operations and Utilities Guide	SC33-1685
CICS Supplied Transactions	SC33-1686
Programming	
CICS Application Programming Guide	SC33-1687
CICS Application Programming Reference	SC33-1688
CICS System Programming Reference	SC33-1689
CICS Front End Programming Interface User's Guide	SC33-1692
CICS C++ OO Class Libraries	SC34-5455
CICS Distributed Transaction Programming Guide	SC33-1691
CICS Business Transaction Services	SC34-5268
Diagnosis	
CICS Problem Determination Guide	GC33-1693
CICS Messages and Codes	GC33-1694
CICS Diagnosis Reference	LY33-6088
CICS Data Areas	LY33-6089
CICS Trace Entries	SC34-5446
CICS Supplementary Data Areas	LY33-6090
Communication	
CICS Intercommunication Guide	SC33-1695
CICS Family: Interproduct Communication	SC33-0824
CICS Family: Communicating from CICS on System/390	SC33-1697
CICS External Interfaces Guide	SC33-1944
CICS Internet Guide	SC34-5445
Special topics	
CICS Recovery and Restart Guide	SC33-1698
CICS Performance Guide	SC33-1699
CICS IMS Database Control Guide	SC33-1700
CICS RACF Security Guide	SC33-1701
CICS Shared Data Tables Guide	SC33-1702
CICS Transaction Affinities Utility Guide	SC33-1777
CICS DB2 Guide	SC33-1939

CICSPlex SM books for CICS Transaction Server for OS/390

General

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

Administration and Management

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

Programming

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

Diagnosis

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

Other CICS books

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

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

Books from related libraries

Books from related libraries include:

IBM CICS for OS/2 Version 3

Installation, SC33-1580
Operation, SC33-1582
Intercommunication, SC33-1583
Problem Determination, SC33-1584

These books are available on the World Wide Web at:

`http://www.hursley.ibm.com/transaction/bookshelf/manuals/os2/`

IBM CICS for MVS/ESA Version 4

Messages and Codes, SC33-1177
Operations and Utilities Guide, SC33-1167
Problem Determination Guide, SC33-1176
Recovery and Restart Guide, SC33-1182

IBM CICS for VSE/ESA Version 2.3

Messages and Codes, GC33-0925
Problem Determination Guide, SC33-0716
Recovery and Restart Guide, SC33-0702

MVS/Enterprise Systems Architecture SP Version 4

Interactive Problem Control System (IPCS): Customization, GC28-1630
Interactive Problem Control System (IPCS): User's Guide, GC28-1631

MVS/Enterprise Systems Architecture SP Version 5

Interactive Problem Control System (IPCS): Customization, GC28-1461
Interactive Problem Control System (IPCS): User's Guide, GC28-1490

NetView Version 2.4

NetView RODM Programming Guide, SC31-7095
NetView MultiSystem Manager Topology Data Model Reference, SV40-0093

CICS Clients

CICS Clients Administration, Version V1.1, SC33-1436
CICS Clients Administation, Version V2.0, SC33-1792: this book is also available on the World Wide Web at:

`http://www.hursley.ibm.com/cics/bookshelf/manuals/clients/v201/`

Summary of changes

This book is based on the previous edition of the *CICSPlex SM Problem Determination* book, GC33-0791-02.

Changes to this book for CICSPlex SM 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)
- 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.

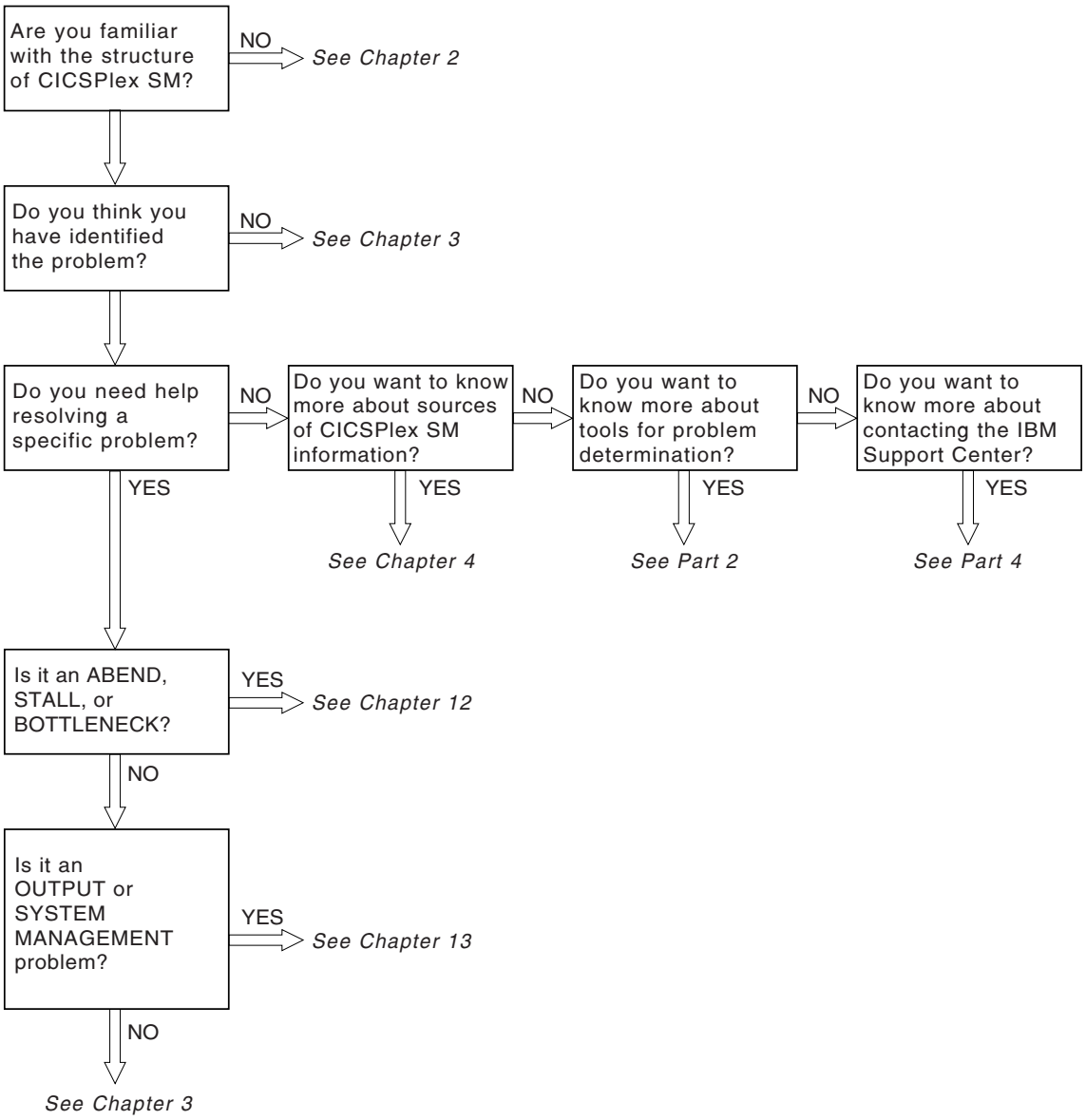

Figure 1. Where to look first

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 an ISPF end-user interface 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/ESA image where a CMAS is run. Just as CMASs are normally connected, so are CASs. In addition, if there is an MVS/ESA 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/ESA 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.

In addition to the visible parts of the CICSPlex SM system that either manifest themselves as MVS/ESA 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/ESA image where a CMAS is run. The ESSS is automatically created when the first CMAS is started in a given MVS/ESA 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/ESA 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.

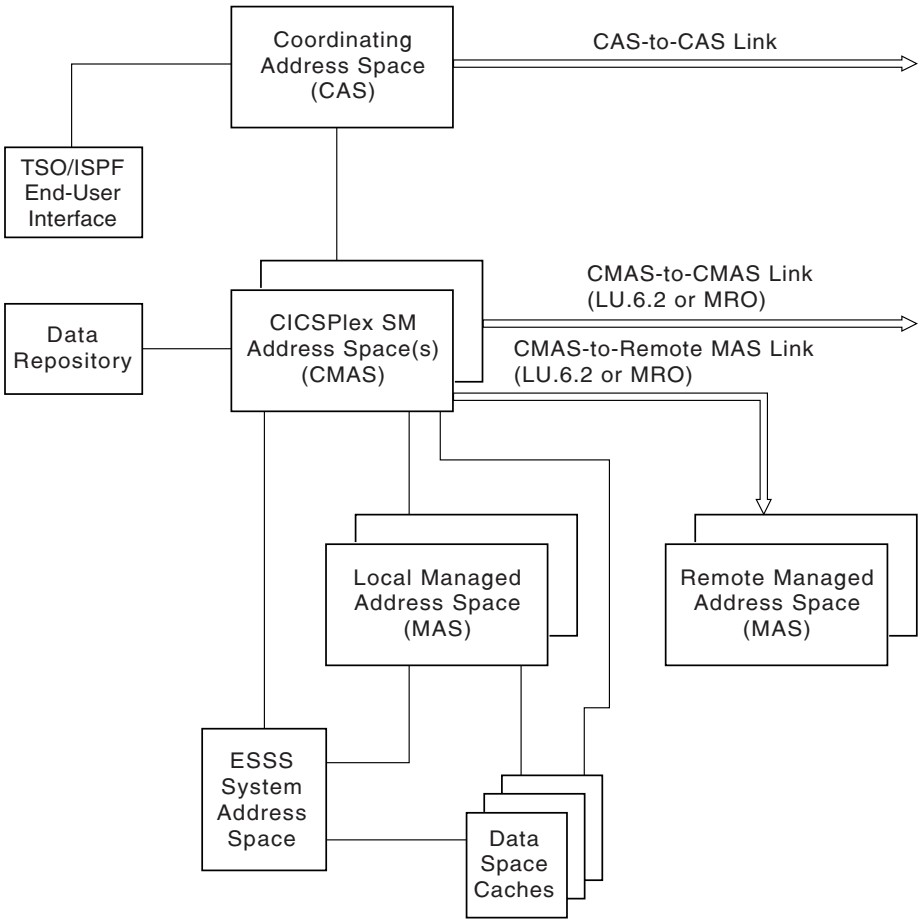

Figure 2. The CICSPlex SM system

The 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 shows a sample CAS network.

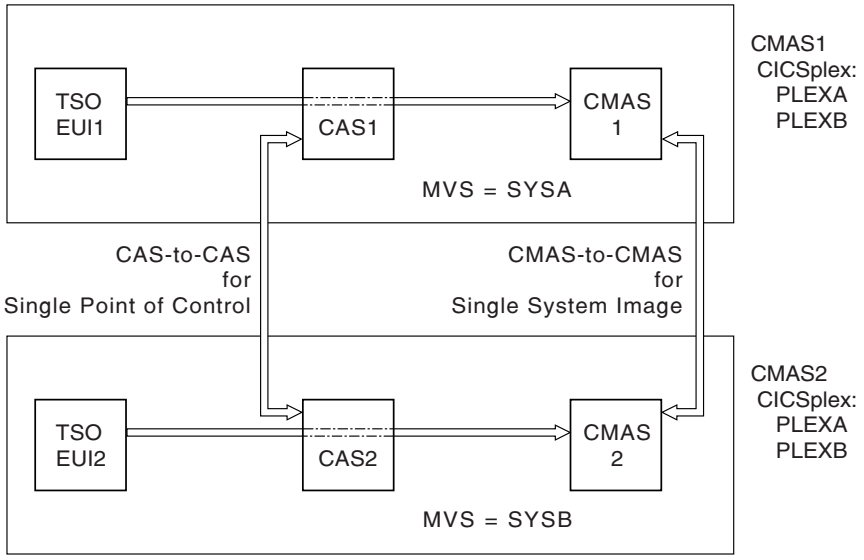

Figure 3. A sample CAS network

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 CICSRGN 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.

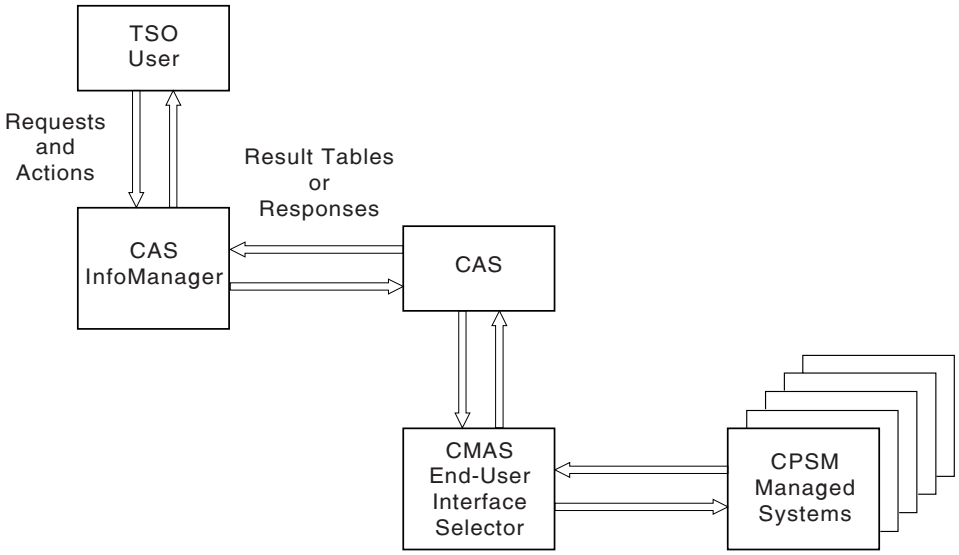

Figure 4. CICSPlex SM end-user interface processing

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.

The CMAS provides the same information about its CICSplexes to the ESSS address space that is running in its MVS/ESA 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 on page 10 shows a sample CMAS network and the service points that result.

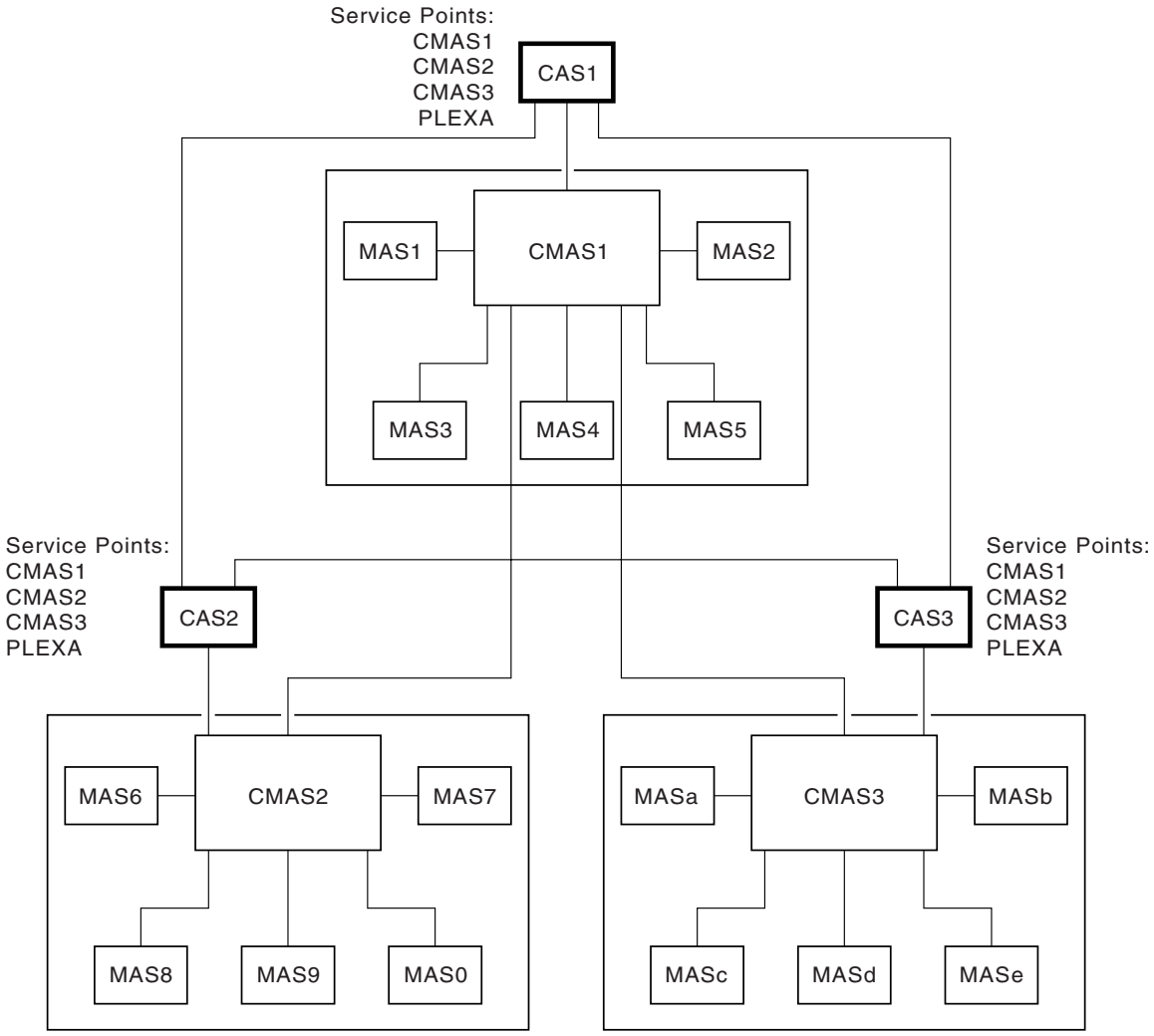

Figure 5. A sample CMAS network

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/ESA 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.

As part of CICS/ESA initialization, it is possible 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/ESA 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 and remote 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/ESA IPL. The ESSS is, in MVS/ESA 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. Additionally, if the CICSPlex SM interface to the NetView Resource Object Data Manager (RODM) is activated, the ESSS notifies NetView when a CMAS agent terminates. Conversely, the ESSS notifies all CMASs reporting to NetView when NetView terminates. 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/ESA 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 163.

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/ESA 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/ESA 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.

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/ESA 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/ESA 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 a CMAS and any remote MAS (CICS/ESA, CICS/MVS, CICS/VSE, or CICS for OS/2), or between a CMAS and another CMAS, can use either CICS intersystem communication (ISC) or interregion communication (IRC) services (usually referred to as multiregion operation, or MRO). Communications provides routines to manage the links between CMASs and remote MASs. 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).

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 OS/390: Program Directory* (or other installation instructions)
- *CICS Transaction Server for OS/390: Installation Guide*

In particular, you might want to try running the installation verification procedures, (IVPs), which are described in *CICS Transaction Server for OS/390: 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?
- Do any of those CICSplexes include remote MASs?

Common types of problem

Refer to Chapter 12, “Abends, stalls, and bottlenecks” on page 129 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 133 if:

- Data in an EUI display is not as expected (entries are missing or incorrect).
- Data in NetView Resource Object Data Manager 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.

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/ESA and CICS)
- Changes to CICS resource definitions
- Maintenance applied to MVS/ESA
- 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 complete list of manuals that may be useful for problem determination, see “CICSPlex SM books for CICS Transaction Server for OS/390” on page xii and “Books from related libraries” on page xii.

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.
- The CICSPlex SM online utility transaction (COLU), described in Chapter 9, “Using the online utility transaction (COLU)” on page 73.
- The CICSPlex SM interactive debugging transactions (COD0 and CODB), described in Chapter 10, “Using the interactive debugging transactions (COD0 and CODB)” on page 79.

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 SM Messages and Codes*. That manual 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® 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 or remote 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 or remote 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).

In CASs, trace data is written to a wrap-around buffer, with new data overwriting old data. Because CAS trace data is not written to any external device, it can be examined only within the context of an address space dump.

CAS tracing operates in two modes, normal and detailed. Normal mode provides a high-level view of the progress of operations over a relatively long period of time. The detailed mode, which operates when extended diagnostic mode (XDM) is turned on, also records nearly all intermodule calls. It provides a very detailed view of operations over a short period of time. Instead of attempting to recover from unexpected events, XDM allows the failing task or process to terminate gracefully.

XDM also generates as much information as possible about the nature of the failure and the environment at the time of the failure.

For details about the tracing facilities provided by CICSPlex SM, see Chapter 5, “Using trace” on page 27. For a description of XDM, see “Extended diagnostic mode” on page 125.

MVS/ESA system dumps

MVS/ESA™ system dumps are an important source of detailed information about problems. For CMASs and local MASs, CICSPlex SM recovery routines produce a system dump when an unexpected error occurs in a supervisory function. Users can also request a system dump at any time.

Whether it is the result of an abend or a user request, a system dump provides an exact image of what was happening in a CICSPlex SM address space at the time the dump was taken. A dump can be used to determine the state of all components in the address space, allowing you to:

- Examine MVS/ESA system trace entries
- Determine subsystem status
- Analyze CAS message trace table entries
- Locate key data areas
- Provide information on all CICSPlex SM components
- Provide information on all active CICSPlex SM tasks
- Provide snapshots of appropriate CICSPlex SM data spaces

Notes:

1. A problem that results in a system dump may actually have been caused by an event that occurred long before the dump occurred. Be sure to review the dump in conjunction with other sources of information, such as logs and traces.
2. If a system dump is not requested a CICS transaction dump will be collected.

For a description of how to use dumps to identify problems in your CICSPlex SM system, see Chapter 6, “Using dumps” on page 39. For a description of the CICSPlex SM IPCS formatting tools, see Chapter 7, “Displaying and formatting dumps with IPCS” on page 49.

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 35.

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 or remote 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.
3. If any local or remote 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, CICS/VSE, or CICS for OS/2. 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 165.
- 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 80.

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.

```
26MAR1999  14:46:30 ----------- INFORMATION DISPLAY ---------------------------
COMMAND  ===> TRACE EYUCMS1A                                SCROLL ===> PAGE
CURR WIN ===> 1         ALT WIN ===>
 W1 =CMAS==============EYUCMS1A=EYUCMS1A=26MAR1999==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.

```
---------------------- 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 2 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 4 5 6 7 8 9 0 1 2
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 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 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 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 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 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 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 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 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 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 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 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 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 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:

Range	Type of trace
0001–1024	Exception trace
1025–2048	Level 1 trace
2049–3072	Level 2 trace
3073–32767	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.

Trace formatting options

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 163.

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 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 EYU9XZUT trace format utility.

```
//jobname  JOB (acct),'name',CLASS=x,MSGCLASS=x
//TRCLST  EXEC PGM=EYU9XZUT,REGION=2048K,PARM='NARROW'
//STEPLIB  DD  DSN=CICSTS13.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

 TASK:00034  METHOD
:XQLK   PRIOR:XQGQ  DEBUG:QLOKEXC   POINTID:     2  TRANID:LP
             MAJOBJ: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 (Part 1 of 2). Example of output from the EYU9XZUT trace format utility

```
   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 (Part 2 of 2). Example of output from the EYU9XZUT trace format utility

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, as described in “BBC display” on page 63.

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:

```
X'152' * X'50' = X'69A0'  (Offset into the table)
X'7F6A5300' + X'69A0' = X'7F6ABCA0'  (Address of most recent entry)
```

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
+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.

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/ESA and CICS, unexpected dumps may be produced at either level.

CICSPlex SM dumps under MVS/ESA

MVS/ESA 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/ESA, 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 11 on page 40 is an example of a CICSPlex SM dump produced under MVS/ESA.

```
BBMZE001E Recovery manager ESTAE entered
BBMZE002E ABEND S00C4-00000011 for ASID=0034-P; 0034/0034/0034,
BBMZE002E CSECT=MXY10 (MXY13+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,GSI=0100041E,MXL04,MXDRX
BBMZE006E ............   MXZRX
BBMZE802I Recovery manager ESTAE 'SDUMP' complete - RC=x'0000'
BBMZE801I Recovery manager ESTAE requested LOGREC recording
BBMZE803I Recovery manager ESTAE requested retry
```

Figure 11. Sample MVS/ESA abend indication and summary

Each MVS/ESA 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=MXY10(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:	`MXY10`
Failing function and offset:	`MXY13 + 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 (local or remote).

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 12 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,CICSMH  ,CICSMH  ,MVSH,CMAS,MCCM,0000041,
            ASRB,EYU0MCCD,08/26/95,09:55:07
+EYUXL0999I XLRV Exiting Successfully
```

Figure 12. 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/ESA jobname
- The name of the CMAS or local MAS (as known to CICSPlex SM)
- The 4-character MVS/ESA 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

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/ESA Jobname
SID	Is the MVS/ESA 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

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/ESA DUMP command
- Using the ISPF end-user interface

Using the MVS/ESA DUMP command

You can issue the MVS/ESA 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 163.

`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/ESA 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.DMDSHTC1,55.QUE1HTC1),END
```

Using the ISPF end-user interface

You can use the SNAP action command on the CICSRGN view to request a system dump for a MAS.

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

1. Issue the CICSRGN view command.
2. Either issue the SNAP primary action command from the COMMAND field, as shown in Figure 13 on page 46, or enter the SNA line action command in the line command field next to EYUMAS1A.

```
26MAR1999 11:30:30 --------- INFORMATION DISPLAY -------------------------------
COMMAND  ===> SNAP EYUMAS1A                                  SCROLL ===> PAGE
CURR WIN ===> 1          ALT WIN ===>
 W1 =CICSRGN=CICSRGN==EYUPLX01==EYUCSG01==26MAR1999==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 13. Requesting a system dump from CICSRGN

3. When the CICS SNAP input panel appears, as shown in Figure 14, 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 14. 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

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/ESA 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/ESA 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.

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.

CICSPlex SM provides two types of IPCS tools:

- A set of panels (driven by a corresponding set of CLISTs) that allow you to display:
  - The data in a coordinating address space (CAS) dump
  - The names and locations of control blocks and areas of a CAS dump
  - Subsystem information
  - Address space-related control blocks
  - Modules loaded by CICSPlex SM
  - Tasks created by CICSPlex SM
  - Storage subpools managed by CICSPlex SM
  - BBC LU 6.2 communication information
- 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/ESA 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 OS/390: Installation Guide*.

Using the CICSPlex SM IPCS panels

You can use the CICSPlex SM IPCS CLISTs to interactively examine the contents of a CAS dump. Many of the CLISTs create IPCS symbol equates. These symbol equates are used by the CLISTs to display storage. You can use the symbol equates as well, while browsing a dump or with IPCS commands, such as CBF. At the end of each panel description, there is a list of the IPCS symbol equates defined by that panel.

Note: Help panels, which can be accessed by pressing PF1, are available for those panels that display the actual dump data.

Accessing the CICSPlex SM panels

As part of setting up the IPCS panels, a DIALOG statement was added to the BLSCECT parmlib member to define the CPSMSSDA component to IPCS. That statement also defined the first panel to appear when the CPSMSSDA component is selected. To display that panel, do the following:

1. From the IPCS primary option menu, select option (2) ANALYSIS.
2. From the IPCS MVS analysis menu, select option (6) COMPONENT.
3. From the IPCS MVS component menu, select CPSMSSDA.

The panel shown in Figure 15 appears.

```
-------------------- CICSPlex SM Subsystem Dump Analysis ----------------------
OPTION  ===>


CAS Subsystem Id  ===>
Base Tech Version ===> 2

   0  SUBSYSTEMS  - Identify CICSPlex SM CAS subsystems
   1  STATUS      - CAS & connected memory status
   2  MESSAGES    - CAS message trace table
   3  COMPONENT   - CAS component level problem analysis

   M  MVS         - Display/format MVS data areas

Enter END command to terminate CICSPlex SM subsystem dump analysis
```

Figure 15. IPCS Subsystem Dump Analysis panel

From this panel you can invoke the underlying CLISTS that find and display diagnostic information from the CAS dump.

Subsystem display

The first thing you need to do is identify the CAS subsystem ID. To do this, select option 0, SUBSYSTEMS, from the Subsystem Dump Analysis panel. This invokes a CLIST which, among other things, creates a table containing one row for each subsystem defined to MVS. An example of a subsystem table appears in Figure 16.

```
--------------------- MVS Subsystem Control Table Display -  ROW 1 TO 13 OF 13
COMMAND ===>                                                  SCROLL ===> CSR

Fmt    Subsystem Name     SSCT       SSVT      Subsystem Contents
Cmd   (Char)   (Hex)     Address    Address    SSCTSUSE  SSCTSUS2     CAS
---   ------  --------   --------   --------   --------  --------   --------
       JES2   D1C5E2F2   00D6D578   00D54760   00D2F5FC  00D54900
       MSTR   D4E2E3D9   00D5CF20   00D5CE08   00000000  00000000
       SMS    E2D4E240   00D6D6D0   00D6D5A0   00000000  00000000
       CICS   C3C9C3E2   00D6DC90   00CEBE70   00D24060  00000000
       CNMP   C3D5D4D7   00D6DDF8   00000000   00000000  00000000
       DB2A   C4C2F2C1   00D5C160   036EDE80   036EDF88  00000000
       DB2B   C4C2F2C2   00D5C138   036EDA80   036EDB88  00000000
       DB2T   C4C2F2E3   00D5C110   036ED680   036ED788  00000000
       OPCA   D6D7C3C1   00D5C0C0   00D49350   00D491F0  00000000
       OPCB   D6D7C3C2   00D5C098   00D490E8   00D4AEA0  00000000
       BBCS   C2C2C3E2   00CE97C8   033DBE98   0301E9D8  00000000     YES
       BBXS   C2C2E7E2   00C84018   00C7C1A0   00C19000  00000000
       IMS.   C9D4E2FF   00F7D000   00000000   00F7D030  00000000
****************************** BOTTOM OF DATA ********************************
```

Figure 16. IPCS Subsystem Control Table Display panel

Each row in this table represents a subsystem defined to MVS. Valid CAS subsystem IDs are denoted by a YES in the CAS column of the table. Find the appropriate CAS subsystem ID and type SD in the Fmt Cmd field to set the default CAS subsystem. In response to the SD command, the CLIST sets up global variables that are used by other CLISTs.

The information in each row is organized as follows:

Subsystem Name
The external name by which the subsystem is known to MVS. This name is provided in both character and hexadecimal format.

SSCT Address
The address of the MVS subsystem CVT that defines the subsystem.

SSVT Address
The address of the service vector table associated with the subsystem.

Subsystem Contents
The contents of the SSCT words reserved for the use by the subsystem.

CAS
Whether or not this subsystem is a full service CAS.

From this panel, the following line commands are available:

SD Sets default CAS subsystem ID.

DC Displays the SSCT (X'1B0' bytes) for the selected subsystem.

DU Displays data pointed to by SSCTSUSE (X'1B0' bytes) for the selected subsystem.

DU2 Displays data pointed to by SSCTSUS2 (X'1B0' bytes) for the selected subsystem.

DV Displays the SSVT (X'1B0' bytes) for the selected subsystem.

Executing this panel also sets up the following IPCS symbol equates:

CVT The MVS CVT.

JESCT The MVS JESCT.

SSCT The default MVS SSCT set by SD line command.

CSCA The CAS subsystem control area set by the SD line command.

CECA0001 The common environment control area set by the SD line command.

Diagnostic overview display

To display an overview of the status of the selected CAS subsystem, select option 1, STATUS, from the Subsystem Dump Analysis panel. This displays a summary of the CAS and all address spaces that were connected to the CAS at the time of the dump. Figure 17 on page 52 is an example of a diagnostic overview display.

```
---------- CICSPlex SM Subsystem Diagnostic Overview (SSID=BB  ROW 1 TO 4 OF 4
COMMAND ===>                                                   SCROLL ===> CSR

CSCA: 0301E9D8     SSCT: 00CE97C8     USERS: 03

          Connected Address Space Summary
          ===============================

Fmt  ADDRESS SPACE IDENTIFICATION    Server    CECA
Cmd    Name      JES Id    ASID       Type    Address
---  --------   --------   ----      ------  --------
     BBCS       STC03991   007A      SYSTEM  00C46D30
     CPSM       STC07729   0026      LOCAL   00AE2D30
     KCBROWN    TSU03209   0082      LOCAL   00A94D30
     TSOUSR2    TSU05810   009A      LOCAL   00A94D30
****************************** BOTTOM OF DATA ********************************
```

Figure 17. IPCS Subsystem Diagnostic Overview panel

This panel shows the CAS subsystem control area (CSCA) and SSCT address of the default CAS subsystem. It also shows how many address spaces are connected and who they are. In Figure 17, there are lines that describe the CAS, the product address space (CPSM), and two TSO users.

Each row is organized as follows:

Address Space Information
The name, JES job ID, and ASID of the address space.

Server Type
Whether the address space contains a SYSTEM, AUXiliary, or LOCAL server structure.

CECA Address
A pointer to the address space's common environment control area (CECA).

Line commands for address spaces

For each address space in the table, the following line commands are available:

F Invokes the VERBX SUMM to format the address space level control blocks for the selected address space.

FA Invokes the IPCS control block formatter to format the ASCB for the selected address space.

DC Displays the address space CECA for X'1B0' bytes for the selected address space.

IPCS symbol equates

The following IPCS symbol equates are defined at this time:

CAXT The address space connect table.

AXTEnn The entry in the address space connect table; there is one for each possible entry.

CECAnn The common environment control area; there is one for each connected address space. In Figure 17, the CPSM address space CECA can be referenced using CECA2.

Message trace table display

The CAS keeps a wrap trace of messages that are issued out of the CAS or any address spaces connected to it. This trace can be used to see such messages as BBMZE002E, BBMZE003E, and BBMZE004E, which describe the SNAP dump information. You can also use this display to locate any unusual messages that might indicate a problem.

The message trace table can be displayed by choosing option 2, MESSAGES, from the Subsystem Dump Analysis panel. A partial subsystem message trace table appears in Figure 18.

```
---------- CICSPlex SM Subsystem Message Trace Table (SSI  ROW 53 TO 78 OF 114
COMMAND ===>                                                SCROLL ===> CSR

ASID  Message Text
-------------------------------------------------------------------------------
007A  BBCSB026I LOGON USER 107 (KCBROWN...JQ....)
007A  BBCSB007I ALLOCATE CONVERSATION 110 (KCBROWN...JQ....) TO SESSION N/A
                (CASC.DEFAULT)
007A  BBCSB008I ALLOCATE SESSION 44 (CASC.DEFAULT) TO CONVERSATION 110
                (KCBROWN...JQ....)
007A  BBCSB007I ALLOCATE CONVERSATION 111 (SP000002.CPSM) TO SESSION 48
                (CASC.DEFAULT)
007A  BBCSB003I ATTACH RECEIVED ON SESSION 48 (CASC.DEFAULT) FOR CONVERSATION
                111 (SP000002.CPSM)
0026  BBMSS201I SIGNON USER KCBROWN ON IG2S1408 AS USERID(KCBROWN)
                GROUP(TSOUSER)
007A  BBCSB029I DEALLOCATE CONVERSATION 110 (KCBROWN...JQ....) ON SESSION 44
                (CASC.DEFAULT)
007A  BBCSB031I CONVERSATION 110: 1207 ELAPSED SECONDS, 18 SENDS (27KB), 14
                RECEIVES (25KB)
007A  BBCSB031I CONVERSATION 111: 1207 ELAPSED SECONDS, 14 SENDS (25KB), 17
                RECEIVES (27KB)
007A  BBCSB015I DEALLOCATED SESSION 48 (CASC.DEFAULT) FROM CONVERSATION 111
                (SP000002.CPSM)
007A  BBCSB015I DEALLOCATED SESSION 44 (CASC.DEFAULT) FROM CONVERSATION N/A
                (N/A.N/A)
007A  BBCSB029I DEALLOCATE CONVERSATION 111 (SP000002.CPSM) ON SESSION N/A
                (CASC.DEFAULT)
0026  BBMSS205I SIGNOFF USER KCBROWN ON IG2S1408
007A  BBCSB027I LOGOFF USER 107 (KCBROWN...JQ....)
```

Figure 18. IPCS Subsystem Message Trace Table panel

Notice that each message is prefixed with the ASID of the address space that issued it. Also remember that this is a wrap trace, so only the last few messages issued are displayed.

This panel also sets the following IPCS symbol equate:

MTT The message wrap trace table in the CAS.

Specific component displays

There are a number of specific component data displays available with the CICSPlex SM IPCS CLISTs. They include:

Program Manager
Responsible for loading modules and invoking tasks.

Storage Manager
Responsible for allocating storage and maintaining subpools.

BBC
Responsible for the LU 6.2 communication between the address spaces and other MVS images.

To get to the component displays, first select option 3, COMPONENTS, from the Subsystem Dump Analysis panel. The panel shown in Figure 19 appears.

```
----------------- CICSPlex SM Subsystem Component Analysis --------------------
OPTION  ===>

Select component to analyze

   1  PROGRAMS    - Locate/display loaded programs
   2  TASKS       - Display execution unit information
   3  STORAGE     - Display storage block/pool information
   4  BBC         - Display communication information


Enter END command to terminate CICSPlex SM subsystem component analysis
```

Figure 19. IPCS Subsystem Component Analysis panel

From this panel, you can select specific component displays.

Load module and programs display

The PGLOADed programs display can be used to either display a list of programs that have been PGLOADed into memory or to determine which, if any, PGLOADed program is associated with a specific storage address. Select option 1, PROGRAMS, from the Subsystem Component Analysis panel and the panel in Figure 20 appears.

```
------------- Program manager - Locate/display PGLOADed program(s) ------------
OPTION  ===>

PROCESS OPTIONS:
   PROGRAM QUEUE:
      TYPE ===> GLOBAL       (Global/Local/All)

   ADDRESS SPACE:            (TYPE=LOCAL requests ONLY)
      NAME ===>              (Default = CAS address space)
      ASID ===>              (Hexadecimal)

SEARCH OPTIONS:
   LMOD NAME  ===> ALL       (Name of module to be located)
   LMOD ADDR  ===>           (Address to be used for search)


Enter END command to return to "Component Analysis" menu
```

Figure 20. IPCS Locate/Display PGLOADed Programs panel

From this panel you can display load modules that have been loaded by the CICSPlex SM PGLOAD service. You can select modules loaded into common storage (GLOBAL), private storage (LOCAL), or both (ALL). You can also select

which address space to display from; the default is the CAS. You can limit the search by specifying the name of a load module to locate or an address from which to begin the search.

The resulting display is presented in tabular form, one row for each PGLOADed load module, as shown in Figure 21.

```
----- Program Manager - PGLOADed Module Display  ----------  ROW 1 TO 26 OF 27
COMMAND ===>                                                   SCROLL ===> CSR

Fmt   Module  Pgm Use  Entry    Load Module    Owner/Free  Module Attributes
Cmd    Name    Q  Cnt Address  Address  Size  ASCB   TCB   RENT FIX FREE RTE
---  -------- --- --- -------- -------- ---- ------ ------ ---- --- ---- ---
     BBM9DE20 GBL  1  02EE6728 02EE6728   1K F5B500 AF19C8 YES  NO  EOM  ---
     BBM9DE08 GBL  2  02E257F0 02E257F0   2K F5B500 AF19C8 YES  NO  EOM  ---
     BBM9KA30 GBL  1  02F0DCD8 02F0D698   2K F5B500 AF19C8 YES  NO  EOM  FVT
     BBC9ZZ10 GBL  1  032AC810 032AC810   2K F85A00 AF2A80 YES  NO  EOM  ---
     BBM9SZ90 GBL  1  03047488 030472D8   3K F85A00 AF2A80 YES  NO  EOM  ---
     BBM9ZZS1 GBL  1  02F93F68 02F91B30  10K F85A00 AF2A80 YES  NO  EOM  ---
     BBM9LY01 GBL  1  02F09478 02F09418  11K F85A00 AF2A80 YES  NO  EOM  ---
     BBM9LZ10 GBL  1  02F9E9D8 02F966E8  35K F85A00 AF2A80 YES  NO  EOM  ---
     BBM9ZE00 GBL  1  032A9F08 032A5358  20K F85A00 AF2A80 YES  NO  EOM  FVT
     BBM9ZZ00 GBL  1  02EF1D90 02EEA320  32K F85A00 AF2A80 YES  NO  EOM  FVT
     BBM9LZ20 GBL  1  00CDFE18 00CDE440  11K F85A00 AF2A80 YES  NO  EOM  ---
```

Figure 21. IPCS PGLOADed Module Display panel

The information is organized as follows:

Module Name
The program (load) module name.

Pgm Q
The program queue on which the program manager PGHD was found.

Use Cnt
How many PGLOADs for the module have been satisfied with the displayed copy of the module.

Entry Address
The load module entry point address.

Load Module
The load module's physical start address and the amount of storage it occupies.

Owner/Free
Pointers to the ASCB and task control block (TCB) for the task that PGLOADed the module.

Module Attributes
Indicates:

- The module's reentrance (RENT) state (YES or NO).
- The module's page fix (FIX) state (YES or NO).
- When the module storage will be released (FREE=EOM or EOT).
- Whether a special run time environment (RTE="C" or FVT) exists for the module.

For each module name, the following line commands are available:

DPH Displays the first X'1B0' bytes of the program header block (PGHD).

DPM Displays the first X'1B0' bytes of the load module.
DPE Displays the first X'1B0' bytes starting at the entry point.
DPT Displays the first X'1B0' bytes of the owner TCB.
DRE Displays the first X'1B0' bytes of the runtime environment.

This panel also sets the following IPCS symbol equates:

GPMQ#load_module_name
There is one equate for each global load module. These equates refer to the beginning of the load module found in global (CSA) storage.

LPMQnnnn#load_module_name
There is one equate for each local load module. The nnnn is a number which uniquely identifies the address space. Typically, for the CAS, it will be 0001. These equates refer to the beginning of the load module found in local (private) storage.

Active task display

The Execution Unit Information display can be used to determine which execution units (created by the PGEXEC service) are active in the specified address space. Selecting option (2), TASKS, from the Subsystem Component Analysis panel displays the screen shown in Figure 22.

```
------------------- Program Manager - Execution Unit Display ------------------
OPTION  ===>

ADDRESS SPACE:
   NAME ===>              (Default = CAS address space)
   ASID ===>              (Hexadecimal)


Enter END command to return to "Component Analysis" menu
```

Figure 22. IPCS Execution Unit Display panel

On this panel, enter either the connected address space name or the ASID, or leave it blank to default to the CAS.

The resulting information is presented in tabular form with one row for each active execution unit, as shown in Figure 23 on page 57.

```
----- Program Manager - Execution Unit Display (BBCS(X'0061  ROW 1 TO 12 OF 12
COMMAND ===>                                                  SCROLL ===> CSR

Fmt  Task    TCB     Exec  Exec  Pgm Name  Parm/CSRB    PXCW    Task term
Cmd   Id   Address   Mode  Key   or GSI #   Address   Contents   ENB ptr
---  ----  --------  ----  ----  --------  ---------  --------  ---------
     S1     A55E88   SUPV   04   BBM9XCL1  80A7F374   7F648AF0  7F659370
     C1     A60840   SUPV   04   BBM9CLIN  8381C88C   00000000  7F7047B8
     C2     A67418   SUPV   04   BBM9MZ00  8381C88C   00000000  7F704938
     S2     A677B8   SUPV   04   BBC9SJ00  FF703DE8   00000000  00000000
     S3     A67C58   SUPV   04   BBC9SI00  FF703DE8   00000000  00000000
     S4     A67E88   SUPV   04   BBC9SD00  FF703DE8   00000000  00000000
     S5     A6E370   SUPV   04   BBC9SC00  FF703DE8   00000000  00000000
     S6     A6E6A0   SUPV   04   BBC9SB00  FF703DE8   00000000  00000000
     C3     A6EC58   SUPV   04   BBC9SA00  8381C88C   00000000  7F706760
     S7     A6EE88   SUPV   04   BBM9SZ31  80A787E4   00000000  00A787D4
     S8     AF2348   SUPV   04   BBM9SZ30  80A787B4   00000000  00A787A4
     C4     AF2678   SUPV   04   BBM9SZ01  8381C88C   00000000  7F7068E0
****************************** BOTTOM OF DATA ********************************
```

Figure 23. IPCS Execution Unit Display panel

The information is organized as follows:

Task Id
The type and initiation order of each execution unit (lowest numbered is the most recently created unit).

C=direct subtask of the server control task
S=component server subtask
A=application task

TCB Address
A pointer to the MVS TCB associated with the unit.

EXEC Mode
The task execution mode.

SUPV=supervisor
PROB=problem

Pgm Name or GSI #
The name of the program (load) module associated with the unit or, for GSI-related units, the GSI route code for the first service entered at PGEXEC time.

Parm/CSRB Address
A pointer to the parameter or GSI service CSRB passed to the first program or service executed under the unit.

PXCW Contents
The program execution control word contents.

Task Term ENB ptr
The address of the EOT ENB provided at PGEXEC time.

For each execution unit, the following line commands are valid:

DXB Displays the first X'1B0' bytes of the program execution block (PGXB).

DPB Displays the first X'1B0' bytes of the related program service Block (PSB).

DXT Displays the first X'1B0' bytes of the related TCB.

DPM Displays the first X'1B0' bytes of the program module.

DPP Displays the first X'1B0' bytes of the program parameter area.

DXW Displays the first X'1B0' bytes of data at the address in PXCW. In the case of a transaction, this will be a transaction object.

This display also sets the following IPCS symbol equate:

PGXBnnnnexecution_id
There is one equate for each execution unit. nnnn is a unique address space identifier, usually 0001 for the CAS. The execution_id is the task ID as displayed on the Execution Unit Display. The equates refer to the address of the PGXB.

Storage displays

From the Storage Manager display, you can display two types of storage:

- Storage blocks, which is storage allocated and maintained via the CAS internal storage allocation component.
- Subpool storage, which is storage allocated and maintained via the CAS internal subpool component.

To display either type of storage, select option 3, STORAGE, from the Subsystem Component Analysis panel. The panel shown in Figure 24 is displayed:

```
-------------- Storage Manager - Locate/display allocated storage -------------
OPTION  ===>

PROCESS OPTIONS:

   STORAGE QUEUE:
      TYPE ===> GLOBAL       (Global/Local)
      POOL ===> NO           (Yes/No)

   ADDRESS SPACE:            (TYPE=LOCAL displays ONLY)
      NAME ===> BBCS         (Default = CAS address space)
      ASID ===> 0061         (Hexadecimal)


SEARCH OPTIONS:

   BLOCK ADDR ===>           (Address of block to be located/displayed)
   POOL ID    ===>           (Hex pool id of pool to be displayed)


Enter END command to terminate storage analysis
```

Figure 24. IPCS Locate/Display Allocated Storage panel

From this panel you can choose to display storage blocks or storage pools (POOL), global or local storage (TYPE), and the address space you are interested in (ADDRESS SPACE). You can also limit the displays by specifying a specific storage block address or a specific storage pool ID.

Storage blocks

The storage block display can be used to determine what local or global storage was allocated at the time the dump was taken. To display this storage, specify either LOCAL or GLOBAL as the storage type and specify NO for pool. Then specify address space information or leave it blank to default to the CAS.

The resulting information is presented in tabular form, with one row for each allocated storage block, as shown in Figure 25.

```
-- GLOBAL Storage  (Total=583K;Pools=282K) ---------------  ROW 1 TO 15 OF 157
COMMAND ===>                                                  SCROLL ===> CSR

   S t o r a g e  B l o c k   Auto-free options           Storage Block
S   Address  Size  SP# Key   Type   ASCB    TCB              Contents
-  -------- ------ --- ---   ----  ------ ------  ----------------------------
   00CFFF68    152 241   4   EOS   F9B280 6F2B80  BBM6GSIB..{.1 ...92..?.......
   00D00FA0   8.9K 241   8   EOS   F9B280 6F2B80  BBM6CTRH..{.1....92..?.......
   00D032B8    144 228   4   EOT   F9B280 6F2B80  BBM6PRVH..U .................
   00D03348  11.8K 228   4   EOS   F9B280 6F2B80  BBM6PGHD..{.U ...92..?.......
   0495EC50   5.0K 228   4   EOT   F34B80 64E928  BBM6SPXB.L ..WK.........N...
   04A61610  26.5K 228   4   EOT   F34B80 64E928  BBM6SPXB.NK-.WI.........W. .
   04A6D238   3.7K 228   4   EOT   F34B80 64E928  BBM6SPXB.L ...VQ.........WK..
   05208040  24.0K 228   4   EOT   F34B80 64E928  BBM6SPXB.LS0...Y............
   0520E5D8   2.7K 228   4   EOT   F34B80 64E928  BBM6SPXB.L ..L...........W..
   0520EEA8  16.6K 228   4   EOT   F34B80 64E928  BBM6SPXB.LS0...-...........Q.
   05213460  11.0K 228   4   EOT   F34B80 64E928  BBM6SPXB.LS0...-............
   05216460  11.0K 228   4   EOT   F34B80 64E928  BBM6SPXB.LS0.L.-............
   05D28590   6.3K 228   4   EOT   F34B80 64E928  BBM6SPXB.T.H.NN.........KE{.
   05D329D0    728 227   4   EOM   F34B80 667E88  BBM6PGHD..{.T ...3....=H.....
   05D32CA8    856 227   4   EOM   F34B80 667E88  BBM6PGHD..{.T ...3....=H.....
```

Figure 25. IPCS Global Storage Display panel

The title line displays the total amount of GLOBAL storage (in KB) occupied by all blocks identified in the table. Each row displays the attributes of one storage block. The information is organized as follows:

Storage Block
The location, size, and MVS subpool and key of the block.

Auto-free options
When the block will be automatically deleted:

End-of-system (EOS)
End-of-memory (EOM)
End-of-task (EOT)

Storage Block Contents
The contents of the first 16 bytes of the block.

For each storage block, the following line command is valid:

S Displays the first X'1B0' bytes of the storage block contents.

Sorting is available from the command line. You can sort on address, size, contents, key, subpool, storage block, and auto-free options. Enter the SORT command, specifying one of the following parameters to identify the data on which you want to sort:

A The address, in descending order.

CO The contents of the storage block, in alphabetical order.

K The MVS storage key, in descending order.

SI The storage block size, in descending order.

S The storage block MVS subpool number, in descending order.

B Information in the storage block, in descending order based on:

- **A** Address
- **C** Storage block contents
- **K** MVS storage key
- **SI** Storage block size
- **S** Storage block MVS subpool number

F The Auto-free options, in descending order based on:

- **A** The owning ASCB
- **T** The owning TCB
- **TY** When the storage pool is to be freed

There are no additional IPCS equates defined at this time.

Cell pool storage

The storage pool display can be used to determine which local or global storage pools existed at the time the dump was taken. To display these storage pools from the Storage Manager panel, specify either LOCAL or GLOBAL as the storage type and specify YES for pool. Then specify address space information or leave it blank to default to the CAS.

The resulting information is presented in tabular form with one row for each allocated storage pool, as shown in Figure 26.

```
----- GLOBAL Pools  (Total=270K) -------------------------  ROW 1 TO 18 OF 18
COMMAND ===>                                                   SCROLL ===> CSR

     Storage Pool Information         Cell Statistics        Auto-free options
S  Pool Id   Size  SP# Key Ext   Size  Total Used Free  HWM  Type  ASCB   TCB
-  -------- ------ --- --- ---  ------ ----- ---- ---- ----  ---- ------ ------
   05E867A0   5.1K 241   4   1     256    20    0   20    1  EOS  F9B280 6F2B80
   05E867B0   5.7K 231   4   1     536    10    0   10    4  EOS  F9B280 6F2B80
   05E867C0   2.5K 241   4   1     168    15    0   15    6  EOS  F9B280 6F2B80
   05E867D0   5.7K 241   4   1     536    10    4    6    4  EOS  F9B280 6F2B80
   05E867E0   8.7K 241   4   1     280    30   21    9   23  EOS  F9B280 6F2B80
   05E867F0   3.9K 241   4   1     104    30    1   29    1  EOS  F9B280 6F2B80
   05E86800   9.2K 228   4   1    2488     4    0    4    0  EOS  F9B280 6F2B80
   05E86810   2.3K 241   4   1     536     5    3    2    3  EOS  F9B280 6F2B80
   05E86820   4.3K 231   4   3     128    37   20   17   30  EOT  F9B280 6F2B80
   05E86830   1.2K 231   4   1     184    10    0   10    6  EOM  F9B280 667C58
   05E86840  13.8K 228   4   2    1344    10    0   10    6  EOT  F34B80 64E928
   05E86860  80.5K 228   4   7    1128    73    0   73   56  EOT  F34B80 64E928
   05E86870    960 228   4   2      96    10    0   10    6  EOT  F34B80 64E928
   05E86880  35.1K 228   4   5    1112    33    0   33   20  EOT  F34B80 64E928
   05E86890   4.6K 228   4   5      48    93    1   92   68  EOT  F34B80 64E928
   05E868A0  72.2K 228   4   5    1112    67    0   67   59  EOT  F34B80 64E928
   05E868B0  14.3K 228   4   8      48   312    1  311  190  EOT  F34B80 64E928
   05E868C0    280 228   4   1      56     5    0    5    0  EOT  F34B80 64E928
****************************** BOTTOM OF DATA ********************************
```

Figure 26. IPCS Global Pools Display panel

The title line displays the total amount of GLOBAL storage (in KB) occupied by all pools in the table. Each row displays the attributes of one storage pool. The information is organized as follows:

Storage Pool Information
The internal pool identification number, total pool size (in KB), MVS subpool number, storage key, and number of allocated pool extents.

Cell Statistics
The size of each storage pool cell, the total number of cells in the pool, the number of cells in use (USED) and available for use (FREE), and the largest number of cells in use at any one time (HWM).

Auto-free options
When the pool will be automatically deleted:

End-of-system (EOS)
End-of-memory (EOM)
End-of-task (EOT)

For each storage pool, the following line commands are valid:

S Displays the pool extent data.

H Displays the first X'1B0' bytes of the storage pool anchor block (SPAB).

Sorting is available from the command line. You can sort on pool ID, pool size, MVS storage key, MVS subpool, storage pool information, and auto-free options. Enter the SORT command, specifying one of the following parameters to identify the data on which you want to sort:

A The pool ID, in descending order.

K The MVS storage key of the storage pool, in descending order.

SI The storage pool size, in descending order.

S The storage pool MVS subpool number, in descending order.

P Information in the storage pool, in descending order based on:

- **E** Number of allocated extents
- **I** Storage pool ID
- **K** MVS storage key of the storage pool
- **SI** Storage pool size
- **S** Storage pool MVS subpool number

C Information under Cell Statistics, in descending order based on:

- **F** Number of free cells in the storage pool
- **H** High water mark
- **S** Cell size
- **T** Total number of cells allocated to the storage pool
- **U** Total number of cells currently in use

F The Auto-free options, in descending order based on:

- **A** The owning ASCB
- **T** The owning TCB
- **TY** When the storage pool is to be freed

There are no additional IPCS equates defined at this time.

The storage pool extent display can be used to determine the location, size, and attributes of each extent allocated to the selected storage pool at the time the dump was taken. To display this table, type an S next to the desired subpool and press Enter. In Figure 26 on page 60, selecting the twelfth subpool will result in a display similar to Figure 27.

```
----- GLOBAL Pool Extents  (Poolid=05E86860) ----------------  ROW 1 TO 7 OF 7
COMMAND ===>                                                   SCROLL ===> CSR

        SPXB              Extent            Cell Locators    1st Cell Content
S  Address  Valid   Size  Cells SPN Key   First      Last   <--- 18 bytes --->
-  -------- -----  ------ ----- --- ---  -------- --------  ------------------
   05D3A9C8  YES     5.4K     5 228  4   05D3A9F8 05D3BB98  ................ST
   05D5B9C8  YES     5.4K     5 228  4   05D5B9F8 05D5CB98  ................IM
   04AB09C8  YES     5.4K     5 228  4   04AB09F8 04AB1B98  ................IM
   04A710F8  YES     7.2K     7 228  4   04A71128 04A72B98  ................ST
   04A6DF58  YES    12.8K    11 228  4   04A6DF88 04A70B98  ................IN
   04A68950  YES    17.3K    16 228  4   04A68980 04A6CB98  ................ST
   04A61610  YES    26.5K    24 228  4   04A61640 04A67B98  ................ST
****************************** BOTTOM OF DATA ********************************
```

Figure 27. IPCS Global Pool Extents Display panel

The title line displays the internal identification number assigned to the pool at the time it was created. Each row displays the attributes of one storage pool extent. The information is organized as follows:

SPXB
The location of the pool manager extent block that describes the extent. The Valid field indicates whether the storage pool extent block (SPXB) passed certain validity checks.

Extent
The extent size (in KB), the number of cells in the extent, and the MVS subpool number and storage key.

Cell Locators
The location of the first and last cell in the extent.

1st Cell Content
The first 18 bytes of the first cell in the extent.

For each storage pool extent, the following commands are valid:

S Displays individual cell header information.

H Displays the first X'1B0' bytes of the SPXB.

There are no additional IPCS equates defined at this time.

The storage pool cell display can be used to determine the location, header (SPFX) validity and allocation status of each cell contained in a storage pool extent at the time the dump was taken. To display this table, type an S next to the desired extent and press Enter. In Figure 27, selecting the third extent results in a display similar to Figure 28 on page 63.

```
----- GLOBAL Pool Cells  (Poolid=05E86860) -----------------  ROW 1 TO 5 OF 5
COMMAND ===>                                                   SCROLL ===> CSR

    Cell     Valid  In           Contents of data portion of cell
S  Address   SPFX   Use  <-------------------- 52 bytes -------------------->
-  --------  -----  ---  ------------------------------------------------------
   04AB09F8   YES   NO   ................IMSCRDR ........X.T.9.H.............
   04AB0E60   YES   NO   ................STARTING........X.V;}E1.............
   04AB12C8   YES   NO   ................STARTING........X.V;BD?.............
   04AB1730   YES   NO   ................STARTING........X.V;..8.............
   04AB1B98   YES   NO   ...........+....STARTING........X.V;..R.............
****************************** BOTTOM OF DATA ********************************
```

Figure 28. IPCS Global Pool Cells Display panel

The title line displays the internal identification number assigned to the pool at the time it was created. Each row displays information about a single cell contained in the extent. The information is organized as follows:

Cell Address
The location of the storage cell prefix (SPFX).

Valid SPFX
Whether the cell prefix passed certain validity checks. A value of NO in this column indicates a probable cell overlay condition.

In Use
Whether the cell was allocated (YES or NO) at the time the dump was taken.

Contents
The first 52 bytes of the user data portion of the cell.

For each cell, the following commands are available:

S Displays the first X'1B0' bytes of the cell contents.
H Displays the first X'1B0' bytes of the SPFX.

There are no additional IPCS equates defined at this time.

BBC display

The BBC LU6.2 Communication display can be used to view pertinent data structures, the internal BBC wrap trace, systems connected with BBC and active sessions. Selecting option 4, BBC, from the Subsystem Component Analysis panel displays the screen shown in Figure 29 on page 64, from which you can display BBC diagnostic information.

```
----------------------- BBC LU6.2 Communication Analysis ----------------------
OPTION  ===>

Select display option

   0  SZV         - Display BBC Server Communication Vector
   1  TRACE       - Display BBC internal trace table
   2  SYSTEMS     - Display systems defined to BBC
   3  SESSIONS    - Display active BBC sessions


Enter END command to terminate BBC LU 6.2 communication analysis
```

Figure 29. IPCS BBC LU6.2 Communication Analysis panel

Selecting option 0, SZV, from this panel and using the IPCS CBF command displays the BBC Server Communication Vector. This data structure contains pointers to all the BBC data structures in addition to current system status.

Selecting option 1, TRACE, from this panel causes the underlying CLIST to find the internal BBC wrap trace in the dump, and if available, list it. To interpret the output from the BBC trace, refer to “Interpreting the LU 6.2 communications trace” on page 36. A partial BBC trace display is shown in Figure 30.

```
IPCS OUTPUT STREAM ----------------------------------------- LINE 0 COLS 1 78
COMMAND ===>                                                SCROLL ===> CSR
****************************** TOP OF DATA ***********************************

BBCTRACE
LIST 7F72B2F0 ASID(X'0011') LENGTH(32016) AREA
ASID(X'0011') ADDRESS(7F72B2F0) KEY(48)
7F72B2F0. 00007D10 00000038 00000188 00000190 |..'........h....|
7F72B300. 00667C58 00000000 00000098 00000038 |..@........q....|
7F72B310. 00150003 0015F402 7F651A40 00000000 |......4.".. ....|
7F72B320. 00000003 00000011 7F5CFF28 7F670F28 |........"*.."...|
7F72B330. 7F5DCA58 0000013D 00000000 00000000 |")..............|
7F72B340 LENGTH(16)==>All bytes contain X'00'
7F72B350. 00667C58 00000000 00001058 00000038 |..@.............|
7F72B360. 00150802 0015F402 7F720E20 00000000 |......4.".......|
7F72B370. 00000002 00000011 7F5CFF28 7F670F28 |........"*.."...|
7F72B380. 00000008 00000000 00000005 000005D7 |...............P|
7F72B390. 00000000 007212FF 00380000 00000000 |................|
7F72B3A0. 00667E88 00000000 00001058 00000038 |..=h............|
7F72B3B0. 00120000 00129322 7F71EC80 00000000 |......l.".......|
7F72B3C0. 00000000 0000013A 00000000 00000005 |................|
7F72B3D0. 000005D7 00000000 007212FF 00380000 |...P............|
7F72B3E0. 00000000 00000000 00000000 0000C1E5 |..............AV|
```

Figure 30. IPCS BBCTRACE Output Stream panel

To display a list of the MVS images defined to BBC at the time of the dump, use option 2, SYSTEMS, from the BBC LU6.2 Communication Analysis panel. The resulting information is presented in tabular form with one row for each system defined to BBC, as shown in Figure 31 on page 65.

```
---------------------------- Systems defined to BBC ---------  ROW 1 TO 2 OF 2
COMMAND ===>                                                   SCROLL ===> CSR

    System                   VTAM                          FLU      MODE
S    Name    Loc SMF  SSID  APPL     MODE    ACTSESS   ADDR     ADDR
--  -------- --- ---- ---- -------- -------- ------- -------- --------
    AVNP     YES  AVNP      CASA              2       7F7294D8 7F728B48
    AVND          AVND      CASB              4       7F727B58 7F7281B8
****************************** BOTTOM OF DATA ********************************
```

Figure 31. IPCS Systems Defined to BBC panel

The information is organized as follows:

System Name
The name of the system as defined at ADD time.

LOC
Whether this is the local system.

SMF
The SMF ID of the system as defined at ADD time.

SSID
The MVS subsystem ID for the system, if provided, at ADD time.

VTAM APPL
The VTAM application name as defined at ADD time.

MODE
The VTAM Mode name if defined at ADD time.

ACTSESS
The number of active sessions with the system.

FLU Address
The CSB_FLU address.

MODE Address
The CSB_MODE address.

For each system, the following line commands are available:

DF Formats the CSB_FLU data structure using the IPCS CBF command.
DM Formats the CSB_MODE data structure using the IPCS CBF command.

To display the active sessions on this system, use option 3, SESSIONS, from the BBC LU6.2 Communication Analysis panel. The resulting information is presented in tabular form with one row for each active session, as shown in Figure 32 on page 66.

```
------------------------- Active BBC LU6.2 Sessions --------  ROW 1 TO 4 OF 4
COMMAND ===>                                                   SCROLL ===> CSR

     Session      Source           Destination     SCB      RCB
 S     ID       System/Userid     System/Userid   Address  Address
---  -------- ----------------- ----------------- -------- --------
     0000018D *.BBMCAS          AVNP.PLEXMGR      7F72A088 7F6512B8
     00000181 *.PLEXMGR         AVNP.BBMCAS       7F651098 7F651B38
     00000015 *.TECGSR          AVND.PLEXMGR      7F729098 7F651918
     00000019 *.PLEXMGR         AVND.TECGSR       7F7292B8 7F72A908
****************************** BOTTOM OF DATA ********************************
```

Figure 32. IPCS Active BBC LU6.2 Sessions panel

The information is organized as follows:

Session ID
The internal ID of the active session.

Source
The system and user IDs of the source of the conversation. An * (asterisk) indicates the local system.

Destination
The system and user IDs of the destination of the conversation.

SCB Address
The CSB_SCB address.

RCB Address
The CSB_RCB address.

For each session, the following line commands are available:

DS Formats the CSB_SCB data structure using the IPCS CBF command.

DR Formats the CSB_RCB data structure using the IPCS CBF command.

DP Locates and formats (using the IPCS CBF command) the CZZ_PAB data structure. Note that this data structure is located in the user address space: if that address space is not in the dump, it is not displayed.

DLC Locates and formats (using the IPCS CBF command) the CUZ_LUCB data structure. Note that this data structure is located in the user address space: if that address space is not in the dump, it is not displayed.

DHS Locates and formats (using the IPCS CBF command) the CSD_HS data structure (half session) associated with the session.

DUB Locates and formats (using the IPCS CBF command) the CSI_UCB data structure (logged on user) associated with the session.

DCB Locates and formats (using the IPCS CBF command) the CSI_CCB data structure (conversations) associated with the logged on user.

Additional displays

From the CICSPlex SM Subsystem Dump Analysis panel, you can access additional MVS information. Selecting M, MVS, enables you to display and format MVS data areas from the panel shown in Figure 33.

```
------------------------ Display/format MVS Data Areas-----------------------
OPTION  ===>

To display information, specify the corresponding option number.

   1  SYMPTOMS    - Display MVS symptom information
   2  TRACE       - Display MVS system trace information
   3  LOGREC      - Display MVS system LOGREC information
   4  SUBSYSTEMS  - Display MVS subsystem information
   5  STORAGE     - Display MVS virtual storage information


Enter END command to terminate MVS data area analysis
```

Figure 33. IPCS Active BBC LU6.2 Sessions panel

Selecting 1, SYMPTOMS, executes VERBEXIT SYMPTOMS TERMINAL NOPRINT, which displays symptoms collected by SDUMP, recovery routines, and IPCS.

Selecting 2, TRACE, executes VERBEXIT TRACE, which displays the MVS system trace.

Selecting 3, LOGREC, executes VERBEXIT LOGDATA, which displays the LOGREC buffer found in the dump.

Selecting 4, SUBSYSTEMS, displays the same subsystem table that option 0 on the Subsystem Dump Analysis panel displays.

Selecting 5, STORAGE, executes VERBEXIT VSMDATA, which displays the MVS virtual storage management (VSM) data areas.

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 EYU9D140, but can also be identified to IPCS as CPSM140 when member CICSTS13.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 CPSM140 '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.

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 163.

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.
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 CPSM140 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 45.

`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 CPSM140 '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 CPSM140 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 CPSM140 '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.

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/ESA image until the next IPL. ESSS implements a formal MVS/ESA 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/ESA 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

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 72.

Dumping data structures (DUMP)

The DUMP option reports on the contents of data structures in both the ESSS and the MVS/ESA 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 140 for CICSPlex SM for CICS Transaction Server for OS/390 Release 3. ALL reports on each version of ESSS that has been created at your enterprise.

SUBSYSTEM
Limits the report to the MVS/ESA 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/ESA 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/ESA subsystem and ESSS data structures.

Reloading broadcast functions (RELOAD)

CICSPlex SM uses two MVS/ESA 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 140 for CICSPlex SM for CICS Transaction Server for OS/390 Release 3.

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

The EYU9XEUT JCL

Figure 34 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(140) EOT
/*
```

Figure 34. 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.

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:

CHE	Data Cache Manager
COM	Communications
KNL	Kernel Linkage
QUE	Queue Manager
SRV	Common Services
TOP	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 35 on page 74 is an example of the report produced by the CACHE keyword.

```
CICSPLEX SM 140 CICS/ESA SNAP Utility For JOB CVMCJBC     02/09/98
                    CPSM 140  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 35. 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 36 on page 75 is an example of the report produced by the ESSSINFO keyword.

```
CICSPLEX SM 140 CICS/ESA SNAP Utility For JOB CVMCJBC     02/09/98
        CPSM 140  Kernel Linkage CICSPlex SM ESSS Connection Information
CMAS Name   Job Name   CICS SYSID   MAS Name   Job Name   CICSPlex Name
 CMAS1JB     CVMCJBC    JWB1
 CMAS1C3     CVMTCC3    HTC3         CVMTCT2    CVMTCT2    PLEX1C1
 CMAS1PP     CVMCPPC    PATC         CSYS1PP    CVMCPPM    PLEX1PP
 CMAS1JF     CVMCJFC    CMJF

CICSPLEX SM 140 CICS/ESA SNAP Utility For JOB CVMCJBC     02/09/98
        CPSM 140  Kernel Linkage CICSPlex SM ESSS Program Information
Program Name   Version   Load Point     Date     Time
 EYU9XESS       110       05A00E78    01/28/98   18.01
 EYUTXEPC       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 140 CICS/ESA SNAP Utility For JOB CVMCJBC     02/09/98
        CPSM 140  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                        0010EFFC  00006014  00000200  00000000
```

Figure 36. 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 37 on page 76 is an example of the report produced by the ALL keyword.

```
CICSPLEX SM 140 CICS/ESA SNAP Utility For JOB CVMCJBC     02/09/98
                  CPSM 140  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 37. 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 38 is an example of the report produced by the LOCKSUM keyword.

```
CICSPLEX SM 140 CICS/ESA SNAP Utility For JOB CVMCJBC     02/09/98
        CPSM 140  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 140 CICS/ESA SNAP Utility For JOB CVMCJBC     02/09/98
        CPSM 140  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 38. Sample LOCKSUM report from COLU

The LOCKSUM report produced by COLU provides information about local locks in 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, CICS/MVS, and CICS/VSE MASs with terminal support. Transaction COD0, with some variations, is also supported in CICS for OS/2 MASs. Any variations in the support for COD0 under CICS for OS/2 are documented throughout this chapter. Transaction CODB is not supported in CICS for OS/2 MASs.

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 80. 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 114. 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. CODB is not supported under CICS for OS/2.

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.

Method-level debugging with COD0

After logging onto CICS, enter the COD0 transaction ID to display the COD0 main menu. Figure 39 shows the COD0 main menu displayed under CICS/ESA, CICS/MVS, and CICS/VSE. Figure 40 on page 81 shows the COD0 main menu displayed under CICS for OS/2.

```
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 remote MAS/MAS/CMAS.
     CALL        Call a CICS transaction or program.
     CAPTURE     Capture an EUI View to print.
     DUMP        Call CODB 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 39. COD0 debugging transaction menu (CICS/ESA, CICS/MVS, CICS/VSE)

```
COD0 CICSPlex/SM for CICS/OS2 Debugger                          Applid
:CICSOS2
CMD=>                                                           Scroll=> PAGE

Welcome to CICSPlex/SM for CICS/OS2 debugger, available commands are:

     DUMP      Dump CPSM resources and memory
     LIST      List CPSM resources and tasks
     HELP      Display help on a command or COMPIDs for CPSM components
               HELP <command> for command help, HELP COMPIDs for components
     PRINT     Prints contents of current display
     TRACE     Set CPSM component trace flags
     TRAP      Set tracing flags for a single method

     EBCDIC    Switch DUMP output to EBCDIC
     ASCII     Switch DUMP output to ASCII
               Note: PF2 will toggle between EBCDIC and ASCII

     END      PF3  Ends current output and refreshes prior command
                   (You may have up to 100 outstanding commands)
     EXIT          Exits the debugger
     UP   n   PF7  Scroll up 'n'/MAX lines (default is one screen)
     DOWN n   PF8  Scroll down 'n'/MAX lines (default is one screen)
     TOP           Position to top of window data
     BOTtom        Position to bottom of window data
     RETRieve PF12 Retrieves a prior command (Last 10 commands are stacked)
     PREV     PF4  During DUMP of Queue/Cache List - Previous record
     NEXT     PF5  During DUMP of Queue/Cache List - Next record
     RECord n      During DUMP of Queue/Cache List - Specified record
```

Figure 40. COD0 debugging transaction menu (CICS for OS/2)

To issue a COD0 debugging command, enter it in the CMD=> input field. 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, CICS/MVS, or CICS/VSE environment:

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

PF keys in effect when COD0 is running under CICS for OS/2 are shown in Figure 40.

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, CICS/MVS and CICS/VSE, 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.

Under CICS for OS/2, the maximum number of nested commands is limited by the amount of available memory.

Issuing commands that alter CICSPlex SM

The following COD0 debugging commands (which are not available in the CICS for OS/2 environment) 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 ALLOC command is not available under CICS for OS/2.

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 163.

CLIST
Displays the Allocate CACHE LIST input panel (shown in Figure 41 on page 83), 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 41. 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 ATTACH command is not available under CICS for OS/2.

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 113.

Unlike the START command, which merely starts a CICS transaction within a CMAS, ATTACH crosses the boundary between a CMAS and a local or remote 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 42 is an example of the display for a completed attached task that ran within a single CMAS or MAS.

```
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:  CVMCTSO1   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/RMAS so all information
       appears in the fields of the MAL.
```

Figure 42. Attached task display for a single CMAS or MAS

Figure 43 on page 85 is an example of the display for a completed attached task that either ran in multiple MASs, or ran multiple times in a CMAS.

```
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:  CVMCTSO1   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/RMAS so a queue of OUT records
       was created.

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

Figure 43. Attached task display for multiple CMASs or MASs

When you press Enter, each of the MALs that ran in each region is reconstructed and displayed individually, as shown in Figure 44.

```
COD0 CICSPlex/SM Debugger
CMD=>                                                          Scroll=> PAGE
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 )
```

Figure 44. 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 CALL command is not available under CICS for OS/2.

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 xxxxxxx is a numeric identifier.

The CAPTURE command is not available under CICS for OS/2.

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® 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. In the CICS for OS/2 environment, you can switch between EBCDIC and ASCII displays using PF2. Some parameters of the DUMP command (though not those available under CICS for OS/2) 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. *Not available under CICS for OS/2*.

@method
Calls CODB with the entry point of the specified CICSPlex SM method. *Not available under CICS for OS/2*.

address [/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. *Available under CICS for OS/2 only*.

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. *Available under CICS for OS/2 only*.

CLIST *token*
Calls CODB with the specified token. The token is entered as two 8-byte hexadecimal character strings. *Not available under CICS for OS/2*.

EIB *compid*
Displays the address of the CICS information block for the first transaction running under the specified component. *Not available under CICS for OS/2*.

EIS *compid*
Displays the address of the CICS storage block for the first transaction running under the specified component. *Not available under CICS for OS/2*.

hexvalue
Assumes the hexadecimal value is an address and enters CODB with ALET=0 and the address specified. *Not available under CICS for OS/2*.

hexvalue hexvalue
Displays the address of the specified ALET (first hexadecimal value) at the specified offset (second hexadecimal value). *Not available under CICS for OS/2*.

MODB *compid*
Displays the address of the MODB for the specified component.

MODD
Displays the address of the MODD. *Not available under CICS for OS/2*.

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. *Not available under CICS for OS/2*.

QUE *token*
Displays the data identified by the specified token. The token is entered as two 8-byte hexadecimal character strings. Under CICS for OS/2 only: 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.

STAKEND *compid*
Displays the address of the last stack for the first transaction running under the specified component. *Not available under CICS for OS/2*.

STAKSTRT *compid*

Displays the address of the initial stack for the first transaction running under the specified component. *Not available under CICS for OS/2*.

XLWA

Displays the CICSPlex SM kernel linkage work area.

Figure 45 shows an example XLWA display.

```
COD0 CICSPlex/SM for CICS/OS2 Debugger                          Applid
:CICSOS2
CMD=>                                                           Scroll=> PAGE
 Dumping ASCII   139C0000  for  X'000000E4'
 139C0000 00000000    4559554E 584C5741 E4000000 43535953  EYUNXLWA....CSYS
 139C0010 00000010    34435720 00303134 30004349 43534F53  4CW .0140.CICSOS
 139C0020 00000020    32200043 49435300 00000000 00000000  2 .CICS........
 139C0030 00000030    00435757 3300504C 45583143 57200000  .CWW3.PLEX1CW ..
 139C0040 00000040    00000000 14010000 FA000000 03000000  ................
 139C0050 00000050    504F532F 32001500 30323031 00000102  POS/2...0201....
 139C0060 00000060    10000000 C0000000 80000000 00000000  ................
 139C0070 00000070    00000000 AAE68745 A18C0000 00000000  .......E........
 139C0080 00000080    00000000 00000000 06000000 D4056813  ..............h.
 139C0090 00000090    00000000 06000000 F0056813 00000000  ..........h.....
 139C00A0 000000a0    00000000 00007213 00007113 00006F13  ......r...q...o.
 139C00B0 000000b0    00007013 00006E13 00006D13 00006C13  ..p...n...m...l.
 139C00C0 000000c0    00006B13 00006913 00000000 00000000  ..k...i.........
 139C00D0 000000d0    00000000 00006A13 00000000 00000000  ......j.........
 139C00E0 000000e0    00000000                             ....
```

Figure 45. An example DUMP XLWA display (CICS for OS/2)

Notes for CICS for OS/2 users:

1. The DUMP display provides an address and length field in the third line of the display (both are highlighted in Figure 45). To reposition the dump output, you can overtype the address field with a valid DUMP address (a hexadecimal address, MODB, MOEB, or XLWA), overtype the length field with an appropriate parameter, and press Enter. For example, to scroll to the MOEB of the DAT component, overtype the address field with MOEB, the length field with DAT, and press Enter.
2. Positioning the cursor within the first byte of the four byte address and either pressing PF9 or entering CSRSEL repositions the dump output to that address. Addresses are displayed according to the hardware architecture. For example, an address of E0009C13 on certain processors is displayed as 139C00E0, so the cursor must be positioned within the 13 to scroll to that address.
3. Issuing DUMP without parameters is invalid.

Notes for CICS/ESA, CICS/MVS, and CICS/VSE 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 EXEC command is not available under CICS for OS/2.

The format of the EXEC command is:

```
EXEC method
```

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 110. For a list of valid responses to this command, see “Running a method” on page 113.

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, CICS/MVS, and CICS/VSE 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. *Not available under CICS for OS/2*.

CACHE
Lists the data caches in use by a local MAS or remote MAS.

CAPTURE
Lists all ISPF end-user interface views currently being captured. *Not available under CICS for OS/2*.

CLIST
Lists the data cache lists in use by a CMAS. *Not available under CICS for OS/2*.

COMM
Lists the two communication MAL queues: one for methods executing via the CMAS, and the other for methods routed to a MAS. *Not available under CICS for OS/2*.

DLL
Lists all CICSPlex SM dynamic link libraries (DLLs). *Available under CICS for OS/2 only*.

METH [*compid***]**
Lists all methods within the specified component. If no component is specified, all methods are listed.

QUE
Lists CICSPlex SM queues. *Available under CICS for OS/2 only*.

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. *Not available under CICS for OS/2*.

STCB [ERRORS]
Lists the server-client control blocks. The ERRORS option provides a description of any errors encountered. *Not available under CICS for OS/2*.

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.

VIEWS [*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. *Not available under CICS for OS/2*.

WAIT *compid*
Lists the CICSPlex SM waits for the specified component. If no component identifier is specified, CICSPlex SM waits for all tasks are listed. *Available under CICS for OS/2 only*.

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 46 shows an example of the LIST ALLOC display.

```
COD0 CICSPlex/SM Debugger
CMD=>                                                          Scroll=> PAGE
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 46. 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 CACHE

Lists the data caches in use by a local MAS or remote MAS. Figure 47 on page 93 shows an example of the LIST CACHE display.

```
COD0 CICSPlex/SM Debugger
CMD=>                                                          Scroll=> PAGE
D=Dump lowest ALET:X'1000'
S Cache    ALET     Low Ofs  High Ofs   HWM Allocated (Hex)
_ DMDSCWW1 01FF001B 00000000 00801000       4,229,376 (00408900)
_ WLM1CWW1 01FF0007 00000000 00400000       3,543,040 (00361000)
_ RTA1CWW1 01010042 00000000 00400000         524,288 (00080000)
_ MON1CWW1 01010041 00000000 00400000         524,288 (00080000)
_ TOP1CWW1 01FF0006 00000000 00400000         655,360 (000A0000)
_ COM1CWW1 01010040 00000000 00400000       1,691,648 (0019D000)
_ MAS1CWW1 01FF0008 00000000 00400000         933,888 (000E4000)
_ DAT1CWW1 0101003F 00000000 00400000       1,527,808 (00175000)
_ QUE1CWW1 01FF0005 00000000 00400000       1,048,576 (00100000)
```

Figure 47. An example LIST CACHE display (CICS/ESA, CICS/MVS, CICS/VSE)

The fields on this display are:

Field	Description
Cache	The name of the data cache.
ALET	The ALET of the data cache.
Low Ofs	The lowest offset allocated, which should always be X'00000000'.
High Ofs	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 48 shows an example of the LIST CACHE display.

```
DBG0 CICSPlex/SM for CICS/OS2 Debugger                      Applid
:CICSOS2
  CMD=>                                                        Scroll=> PAGE
  _ Token             ElemLen Keylen Keyoff Records MaxRecs FreeRec    Storage
  _ 00000001-0123FA2C     300      5      0       3      10       1        760
```

Figure 48. An example LIST CACHE display (CICS for OS/2)

The fields on this display are:

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 49 shows an example of the LIST CAPTURE display.

```
COD0 CICSPlex/SM Debugger
CMD=>                                                          Scroll=> PAGE
Enter END to exit.
ViewName  User ID  Count  Debug  AltDebug  Desc
FILES     CVMCW2       2  OT     NT        File Summary
LOCFILE   CVMCW2       2  O8     N8        Local Files
```

Figure 49. An example LIST CAPTURE display

The fields on the LIST CAPTURE display are:

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 50 shows an example of the LIST CLIST display.

```
COD0 CICSPlex/SM Debugger
CMD=>                                                          Scroll=> PAGE
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 50. An example LIST CLIST display

The fields on this display are:

Field	Description
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 51 shows an example of the LIST COMM display.

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

Figure 51. An example LIST COMM display

The fields on this display are:

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/ESA image.
	Remote MAS	From a MAS in another MVS/ESA or VSE/ESA® image.

LIST DLL

Lists all CICSPlex SM dynamic link libraries. Figure 52 shows an example of the LIST DLL display. LIST DLL is available under CICS for OS/2 only.

```
COD0 CICSPlex/SM for CICS/OS2 Debugger                    Applid
:CICSOS2
CMD=>                                                      Scroll=> PAGE
 Cmp StartAdr Length -- Decimal Entry    LastMeth Hndl
 KNL 140A0000 00015183    86403 140A0000 140A8AE4 07A3
      C:\CPSM140\BIN\EYU9XL05.DLL (458752 bytes)
      Last Written:04/10/15 12:46:09
 TRC 14010000 0000D783    55171 14010000 140112D8 08AD
      C:\CPSM140\BIN\EYU9XZ05.DLL (262144 bytes)
      Last Written:04/10/15 12:52:00
 MSG 13FC0000 0000DFF3    57331 13FC0000 13FC1B48 08BA
      C:\CPSM140\BIN\EYU9XM05.DLL (294912 bytes)
      Last Written:04/10/15 12:50:03
 SRV 13EF0000 0000F1E3    61923 13EF0000 13EF2D38 0B2A
      C:\CPSM140\BIN\EYU9XS05.DLL (360448 bytes)
      Last Written:04/10/15 12:50:13
 CHE 13EA0000 000119B7    72119 13EA0000 13EA54C0 07C9
      C:\CPSM140\BIN\EYU9XC05.DLL (409600 bytes)
      Last Written:04/10/15 12:49:12
 DAT 13E50000 0000441F    17439 13E50000 13E50248 0757
      C:\CPSM140\BIN\EYU9XD05.DLL (49152 bytes)
      Last Written:04/06/15 12:05:04
```

Figure 52. An example LIST DLL display

Three lines of information are displayed for each DLL in the list. The first line provides the following fields:

Field	Description
StartAdr	The starting address of the DLL in memory.
Length	The length of the DLL in memory, displayed in hexadecimal.
Decimal	The length of the DLL in memory, displayed in decimal.
Entry	The entry point of the DLL.
LastMeth	The address of the last method in the DLL.
Hndl	The OS/2® module handle.

The second line provides the location from which the DLL was loaded and the size of the DLL, in the form of *directory\filename (nnn bytes)*.

The third line provides one of the following:

Created	The date and time the DDL was created (for HPFS file systems only).
Last Accessed	The date and time the DDL was last accessed (for HPFS file systems only).
Last Written	The date and time the DDL was last written (for HPFS or FAT file systems).

LIST METH

Lists all methods within the specified component. Figure 53 shows an example of the LIST METH display typical of those produced under CICS/ESA, CICS/MVS, and CICS/VSE. Figure 54 on page 98 shows an example of the LIST METH display typical of those produced under CICS for OS/2.

```
COD0 CICSPlex/SM Debugger
CMD=>                                                        Scroll=> PAGE
L=Dump Load Pt  END=Exit.
S Typ Meth Function Fmt Tran LoadPt   ServLevl Assembly Date  Status
_ PUB CWAA ADDTMED  01       077130C0 CPSM140  05/19/98 06.05 ACTIVE
_ PUB CWAB BROTMED  02       07713348 CPSM140  05/19/98 06.05 ACTIVE
_ PUB CWAD DELTMED  03       077138F0 CPSM140  05/19/98 06.05 ACTIVE
_ PUB CWAU UPDTMED  04       07713B40 CPSM140  05/19/98 06.05 ACTIVE
_
```

Figure 53. An example LIST METH display (CICS/ESA, CICS/MVS, CICS/VSE)

The fields on this display are:

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 load detected errors and the method cannot be run.
	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.

```
COD0 CICSPlex/SM for CICS/OS2 Debugger                    Applid
:CICSOS2
CMD=>                                                      Scroll=> PAGE
 _ Typ Cmp Meth Address  EndAddr  Xltd     Xlfd     Tran Flags
 Component:Kernel Linkage
 _ PRV KNL XLBI 140A78B8 140A7DFB 140D0558 140D0574      ACTIVE
 _ PRV KNL XLBV 140A7DFC 140A8AE3 140D05C8 140D05E4      ACTIVE
 _ PRV KNL XLES 140A6A8C 140A78B7 140D04DC 140D04F8      ACTIVE
 _ PUB KNL XLNE        0        0 140D02BC 140D02D8      LOCK,NOTFND
 _ PUB KNL XLPI 140A07E0 140A1DCB 140D007C 140D0098      ACTIVE
 _ PUB KNL XLPQ 140A39B0 140A4C8B 140D01D4 140D01F0      ACTIVE
 _ PUB KNL XLPS 140A1DCC 140A39AF 140D00F8 140D0114      ACTIVE
 _ PUB KNL XLSE 140A04E4 140A07DF 140D0000 140D001C      ACTIVE
 _ PUB KNL XLTP 140A4C8C 140A56B3 140D035C 140D0378      ACTIVE
 _ PUB KNL XLTT 140A56B4 140A6A8B 140D0408 140D0424      ACTIVE
 _
```

Figure 54. An example LIST METH display (CICS for OS/2)

The fields on this display are:

Field	Description
Typ	Indicates that the method is either private (PRI) or public (PUB).
Cmp	The component of the method.
Meth	The name of the method.
Address	The address of the method entry in memory.
EndAddr	The ending address of the method in memory.
Xltd	The transport services definition header for the method.
Xlfd	The transport services field definition array for the method.
Tran	If the method runs asynchronously, the CICS transaction ID used.
Flags	The status of the method as listed under Status on page 97.

LIST QUE

Lists CICSPlex SM queues. The LIST QUE command is available under CICS for OS/2 only. Figure 55 shows an example LIST QUE display.

```
DBG0 CICSPlex/SM for CICS/OS2 Debugger                    Applid
:CICSOS2
CMD=>                                                      Scroll=> PAGE
 _ Token             Type    Records   MaxSz LastRead Blocks   TotSize
 _ 13580014-00000001 WORK          1     392        0      1       392
 _ 1358002C-00000002 WORK          2    4088        0      1      7084
 _ 13580044-00000003 WORK          0       0        0      1         0
 _ 1358005C-00000004 WORK          1    3112        0      1      3112
 _ 13580074-00000005 WORK          1    1160        0      1      1160
 _ 1358008C-00000006 WORK          1     112        0      1       112
 _ 135800A4-00000007 WORK          5    1472        0      1      2712
```

Figure 55. An example LIST QUE display

The fields on this display are:

Field	Description
Token	The hexadecimal queue token.
Type	The type of queue as either WORK or WORKDEL.
Records	The number of records in the queue.
MaxSz	The maximum record length.
LastRead	The number of the record that was last retrieved.
Blocks	The number of blocks allocated for this queue.
TotSize	The total memory used, including any overhead.

You can enter the following in the selection field:

Command	Description
D	Dumps the data in the queue. This is the same as entering DUMP QUE *token* on the command line.

LIST START

Lists the status of all methods you've started or attached. Figure 56 shows an example of the LIST START display.

```
COD0 CICSPlex/SM Debugger
CMD=>                                                          Scroll=> PAGE
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 56. An example LIST START display

The fields on this display are:

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 57 shows an example of the LIST STCB display.

```
COD0 CICSPlex/SM Debugger
CMD=>                                                        Scroll=> PAGE
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 57. An example LIST STCB display

The fields on this display are:

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 58 shows an example of the LIST TASK display typical of those produced under CICS/ESA, CICS/MVS, and CICS/VSE. Figure 59 on page 102 shows an example of the LIST TASK display typical of those produced under CICS for OS/2.

```
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 58. An example LIST TASK display (CICS/ESA, CICS/MVS, CICS/VSE)

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).

```
DBG0 CICSPlex/SM for CICS/OS2 Debugger                          Applid
:CICSOS2
CMD=>                                                           Scroll=> PAGE
  _ Meth Task Aoeb     Local Storage TaskStg  Mal      Flag EIB      Tran
 Component:KNL Pid:805(325) Parent:800(320) Tid:2(2) Term:N/A
  _ XLEV 0006 13C20018 00000000 0000 13C405A8 00000000 0002 000311B0 COEX
  _ NLRT 0007 13C20094 13A30000 07EC 13C405A8 140F00BC 0000 000311B0 COFX
```

Figure 59. An example LIST TASK display (CICS for OS/2)

The fields on this display are:

Field	Description
Meth	The name of the method.
Task	The CICS task number in decimal.
Aoeb	The agent object environment block.
Locl Storage	The address and hexadecimal length of the local work area for this instance of the method.
TaskStg	The task related storage area.
Mal	Address of the MAL.
Flag	The flags for the method.
EIB	The address of the CICS EIB for the task.
Tran	The CICS/OS2 transaction id.
Component	The component id for the CICS task.

Pid	The id of the OS/2 process, in decimal with the hexadecimal value shown in parenthesis.
Parent	The Pid of the parent OS/2 process.
Tid	The id of the OS/2 thread, in decimal with the hexadecimal value shown in parenthesis.
Term	The terminal on which the transaction is running, or N/A if there is no terminal associated with the transaction.

You can enter the following in the selection field:

Command	Description
A	Dumps the Aeob.
L	Dumps local storage.
S	Dumps task storage.
T	Terminates processing (post termination ECB).

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 AAAAAAAA
```

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. `AAAAAAAA` is the address of the invalid stack frame.

```
OPB chain error at AAAAAAAA
```

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. `AAAAAAAA` is the address of the OPB that was next after the previously displayed OPB.

```
Eyecatcher failed for CSFM at AAAAAAAA
```

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

```
DFHEIBLK block invalid at AAAAAAAA, OPB at AAAAAAAA
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 60 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 CICSPlex/SM Debugger
CMD=>                                                        Scroll=> PAGE
Enter END to exit.
View      DistName Type     Class Name    Desc
ACTNDEF   EYUZEGP0 TRTAA    ActDef        Action Definitions
ADMANLS   EYUZEAZ4 TMENU3   ViewTable     Real Time Analysis Admin Views
ADMCONFG  EYUZEAZ1 TMENU3   ViewTable     CMAS Configuration Admin Views
ADMIN     EYUZE$MM TMENU2   ViewTable     ============================
ADMMON    EYUZEAZ2 TMENU3   ViewTable     Monitor Admin Views
ADMTOPOL  EYUZEAZ5 TMENU3   ViewTable     Topology Admin Views
ADMWRKLD  EYUZEAZ3 TMENU3   ViewTable     Workload Manager Admin Views
AIMODEL   EYUZEK50 TOPERATE CicsAiModel   Auto Install Models
AIMODELS  EYUZEK5S TOPERATE CicsAiModel   Auto Install Model Summary
ANALYSIS  EYUZEAZR TMENU1   ViewTable     Real Time Analysis Views
```

Figure 60. 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.

LIST WAIT

Lists the CICSPlex SM waits for one or more components. The LIST WAIT command is available under CICS for OS/2 only. Figure 61 shows an example of the LIST WAIT display.

```
COD0 CICSPlex/SM for CICS/OS2 Debugger                        Applid
:CICSOS2
CMD=>                                                         Scroll=> PAGE
 _ XSWC-EcbAddr  NumPost Created Ecb Hnd  Tmr Hnd  Semiphore
 Object process block for KNL at 13680000, CICS task 6
 _    1 13630018       0  XLEV    800100A4  00000000 \SEM32\EYU\6\136805B8
 _      NO WAITS       0  XLEV    800100A5  00000000 \SEM32\EYU\6\136805D4
 _    5 13630048       0  XLEV    800100A6  00000000 \SEM32\EYU\6\136805F0
 _    2 13630024       0  NLRT    800100A7  00000000 \SEM32\EYU\6\1368060C
 _    3 13630030       0  NLRT    800100A8  00000000 \SEM32\EYU\6\13680628
 _    4 1363003C       0  NLRT    800100A9  00000000 \SEM32\EYU\6\13680644
```

Figure 61. An example LIST WAIT display

The fields on this display are:

Field	Description
XSWC	The XSWC is a method that waits on ECBs. This field is the sequence of the ECB in the XSWC.
EcbAddr	The address of the ECB in the caller's storage.
NumPost	The number of OS/2 DosPostEventSemaphores.
Created	The name of the method that created the ECB.
Ecb Hnd	The OS/2 semaphore handle.
Tmr Hnd	The OS/2 AsyncTimer handle, if the ECB has a timer associated with it.
Semaphore	The name of the OS/2 semaphore.

You can enter the following in the selection field:

Command	Description
P	Posts the ECB.

POST (posting an ECB)

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

The POST command is not available under CICS for OS/2.

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/MVS, CICS/VSE)

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 xxxxxxx is a numeric identifier.

This version of the PRINT command is not available under CICS for OS/2.

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.

STAKSTRT *compid*
Prints the first stack for the specified component.

XLWA
Prints the XLWA.

PRINT (printing current window contents under CICS for OS/2)

The PRINT command prints the current window contents to a printer or file.

This version of the PRINT command is available under CICS for OS/2 only.

The format of the PRINT command is:

```
PRINT [filename] [FF|PAGE]
```

where:

filename
Identifies a printer (such as LPT1, LPT2, or LPT3) or a disk file. Printer names must not include colons (for example, specify LPT1, not LPT1:). If the disk file already exists, the text is appended to the end of the file. If no file name is specified, output is sent to the default printer.

FF|PAGE
FF causes a form-feed character to be inserted before the output. PAGE causes a form-feed character to be inserted both before and after the output.

If you are printing a DUMP screen, only the current screen contents are printed, not the entire range of memory you have selected. You must page down (PF8 or DOWN) and reenter the PRINT command to print more than one screen of data.

On entry to COD0, the default printer is PRN (LPT1). To change the default printer, use the following command:

```
PRINT FILE filename
```

where:

filename
Identifies the new default printer.

PURGE (purging an allocated resource)

The PURGE command purges an allocated resource.

The PURGE command is not available under CICS for OS/2.

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 START command is not available under CICS for OS/2.

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 110. For a list of valid responses to this command, see “Running a method” on page 113.

Figure 62 on page 108 shows an example of the START display.

```
COD0 CICSPlex/SM Debugger
CMD=>                                                          Scroll=> PAGE
Overtype 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 62. An example START display

The START command starts a CICS task that eventually executes method DBG2. 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|OFF|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). *Not available under CICS for OS/2*.

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

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. *Not available under CICS for OS/2*.

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).

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:

Uxxxxxxxx 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 63.

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

Figure 63. Sample formatted MAL display

As shown in Figure 63, 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 3 shows the input allowed for each field type.

Table 3. 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
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 82.)

If the field contains subfields, as described in “Subfields,” 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 64 on page 113.

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

IN
```

Figure 64. Using ALLOC to allocate a resource

Figure 65 shows the allocated resource, /workara, being used as input to a MAL.

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

IN
   *ENM FUNCTION( TEST )
    CHR DEBUG(          )
    EPT OUT_ADDR(A= /workara   O=         )
```

Figure 65. Using an allocated resource in a MAL

The COD0 debugging transaction places the ALET of the allocated storage area 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 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, which is not supported in the CICS for OS/2 environment, 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 66. (CODB can also be entered from the DUMP command of the COD0 transaction.)

```
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

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 66. CODB debugging transaction menu

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

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 163.

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.

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 SM Messages and Codes* book.

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 67 is a sample CODB 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  FFFFF34E  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  YUXEEYURXELS....
00077408  000000A0  009AFC38  03C6B150  0093D140  0093D154  .....F.&;lJ .lJ.
00077418  000000B0  00FAB580  006C1258  A458C562  09D66631  .....%..u.E..O..
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..O...%......
000774C8  00000160  000000D4  00000002  E2E8E2C3  C3E5D4C3  ...M....SYSCCVMC
000774D8  00000170  E3E2D6F1  C3E5D4C3  E6404040  C3E6E6F1  TSO1CVMCW   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=BOTM  P3=END  P4=PREV  P5=NEXT  P6=TOKEN  P7=BACK  P8=FRWD
P9=JUMP  P10=DSJUMP  P11=ALTER  P12=ALET/OFFSET
```

Figure 67. Sample CODB memory display

CODB 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.

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 4 on page 122 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
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 68.

```
26MAR1999  16:19:59 ----------- INFORMATION DISPLAY ---------------------------
COMMAND  ===>                                                SCROLL ===> PAGE
CURR WIN ===> 1        ALT WIN ===>
 W1 =DIAGMSG==========EYUA=====*========26MAR1999==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 68. The DIAGMSG view

Action commands

Table 4 lists the action commands for the DIAGMSG view.

Table 4. 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 OS/390: 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 BBMSS201I 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 69. 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 70 on page 124 shows an example of the DIAGSESS view.

```
26MAR1999  16:19:45 ----------- INFORMATION DISPLAY ---------------------------
COMMAND  ===>                                                SCROLL ===> PAGE
CURR WIN ===> 1        ALT WIN ===>
 W1 =DIAGSESS==========EYUA=====*========26MAR1999==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 70. 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 123, replacing DIAGSESS with DIAGSYS.

Figure 71 is an example of the DIAGSYS view.

```
26MAR1999  16:19:59 ----------- INFORMATION DISPLAY ---------------------------
COMMAND  ===>                                                SCROLL ===> PAGE
CURR WIN ===> 1        ALT WIN ===>
 W1 =DIAGSYS==========EYUA=====*========26MAR1999==16:19:59=PLEXMGR=======2===
CMD SYSTEM   C SMF  SUBS APPLICAT MODE     ACTSESS
--- -------- - ---- ---- -------- -------- --------
    EYUA     *      EYUA EYUCAS1A                2
    EYUB            EYUB EYUCAS1B                2
```

Figure 71. 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 CSRB 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 CSRB 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

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

Because CICSPlex SM has a presence in two major parts of your environment (MVS/ESA and CICS) abends can occur in either place. CAS, ISPF end-user interface, and some CMAS abends occur under MVS/ESA. 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/ESA 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. (SDUMPs are not produced when abends occur in remote MASs.) 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?
- 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 129.

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:

```
26MAR1999  07:02:28 ----------- INFORMATION DISPLAY ---------------------------
COMMAND  ===>                                                  SCROLL ===>
CURR WIN ===> 1        ALT WIN ===>
 W1 =CICSRGN==========EYUPLX01=EYUCSG01=26MAR1999==07:01:51=CPSM==============
BBMXBD15I There is no data that satisfies your request
```

Figure 72. Example of incomplete data returned

In the example shown in Figure 72, 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.

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 JESMSGLG 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 CICSSYS 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 JESMSGLG of the MAS when the NAME parameter is incorrect:

```
DFHSI1517 EYUMAS1A Control is being given to CICS.
EYUXL0003I EYUMAS1A  CPSM Version 140 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 73. Example of JESMSGLG 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 74.

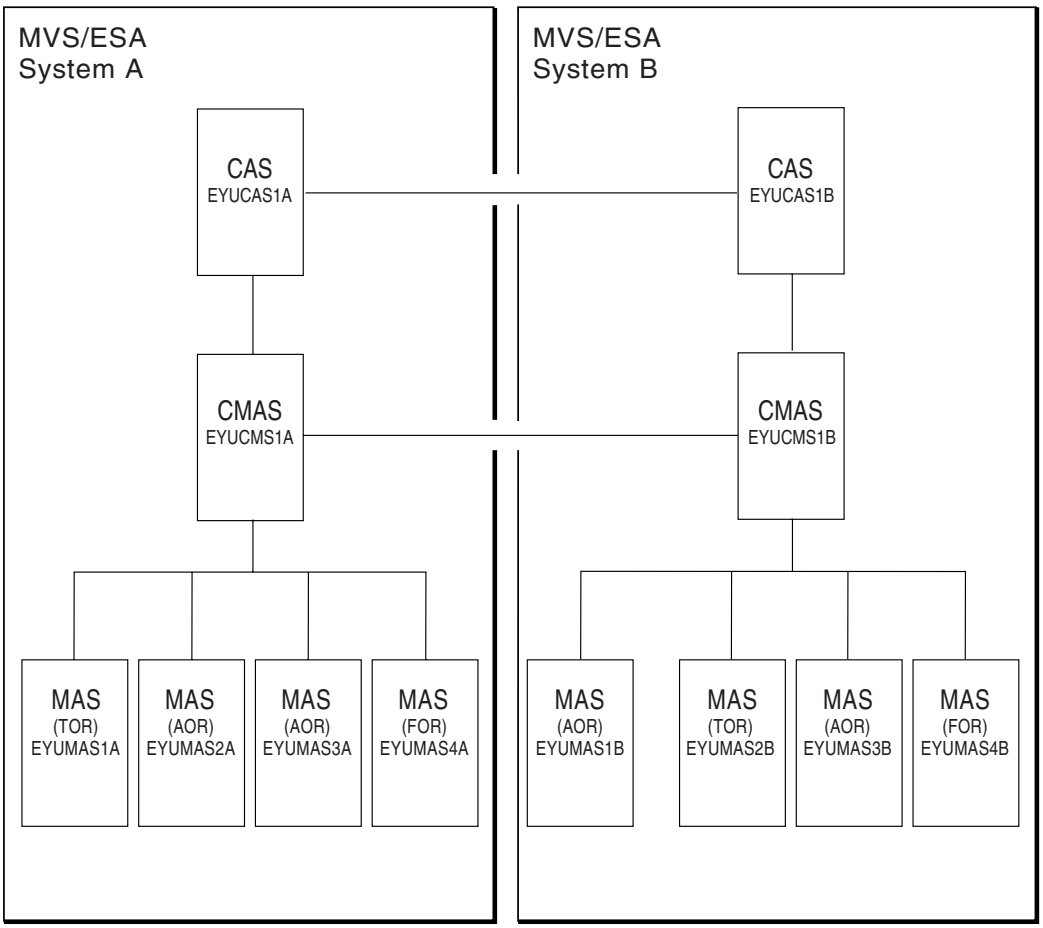

Figure 74. Example of a CICSplex managed by multiple CMASs

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:

```
26MAR1999  06:57:38 ----------- INFORMATION DISPLAY ----------------------------
COMMAND  ===>                                                  SCROLL ===>
CURR WIN ===> 1        ALT WIN ===>
>W1 =MAS===============EYUPLX01=EYUPLX01=26MAR1999==06:57:37=CPSM====8=========
CMD Name     Type   CMAS     Status   MON RTA WLM Description
--- -------- ------ -------- -------- Act Act Act ------------------------------
    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 75. Example of failed connectivity between CMASs

The result shown in Figure 75 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 76 shows the result:

```
26MAR1999  07:41:08 ----------- INFORMATION DISPLAY ---------------------------
COMMAND  ===>                                                 SCROLL ===>
CURR WIN ===> 1        ALT WIN ===>
 W1 =CICSPLEX==========EYUCMS1A=EYUPLX01=26MAR1999==07:40:42=CPSM====2=========
CMD CICSPlex CMAS     Status   MP  Access   Transit  Transit
--- -------- -------- -------- --- Type---- --CMAS-- -Count-
    EYUPLX01 EYUCMS1A ACTIVE   YES LOCAL                   0
    EYUPLX01 EYUCMS1B INACTIVE NO  N/A                     0
```

Figure 76. 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:

```
26MAR1999  15:38:19 ----------- INFORMATION DISPLAY ---------------------------
COMMAND  ===>                                                 SCROLL ===>
CURR WIN ===> 1        ALT WIN ===>
 W1 =MNTRATDQ==========EYUPLX01=EYUMAS1A=26MAR1999==15:38:19=CPSM==============
BBMXBD15I There is no data that satisfies your request
```

Figure 77. 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.

```
26MAR1999  15:48:22 ----------- INFORMATION DISPLAY ---------------------------
COMMAND  ===>                                                  SCROLL ===>
CURR WIN ===> 1        ALT WIN ===>
 W1 =MONACTV==========EYUPLX01=EYUMAS1A=26MAR1999==15:48:22=CPSM====7=========
CMD Def      CICS     Status     Active   Resource Resource Include RODM
--- Name---- System-- ---------- Period-- Name---- Type---  ------- Pop-
    *        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 78. 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 79 on page 138.

```
---------------------------------- 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 79. Example of the monitor section of the MAS view

The view shown in Figure 79 shows that monitoring is active for all resource types except transient data queues.

Incorrect or missing status data in NetView RODM

Use the following check lists to help determine why you might be missing status data or receiving incorrect status data in the NetView Resource Object Data Manager.

Review online definitions

- Issue the MONACTV command. If the monitor definition for the resource in question is not displayed in the MONACTV view, first review “Missing monitor data” on page 136.
- If the MONACTV view shows a status of PENDING for the resource, note the Active Period value. Issue the PERIODEF command, locate the period definition specified in the Active Period field of the MONACTV view, and modify the Start Time and End Time.
- If the monitor definition is active, ensure the RODM Pop field indicates YES. If not, position the cursor on the name of the monitor definition and press Enter to hyperlink to the MONDEF view. Issue the UPD action command to display the Update Monitor definition panel. Change the RODM Status field to YES, as shown in Figure 80 on page 139 and press Enter to redisplay the MONDEF view.

```
------------------------ Update Monitor Definition for EYUPLX01 ---------------
COMMAND  ===>

Definition Name      HTTRAN
Description     ===> Monitor Q* transactions

Resource Definition:
 Name           ===> Q*     Resource Name
 Type           ===> MTRAN  MCONN/MFILE/MJRNL/MPROG/MTDQS/MTERM/MTRAN
 Include        ===> NO     Include for monitoring (NO, YES)
 RODM Status    ===> YES    Operational status for RODM (NO, YES)


Press Enter to update Monitor Definition.
Type END or CANCEL to cancel without updating.
```

Figure 80. Update Monitor Definition RODM Status field

Issue the INS action command to install the modified definition in the correct scope.

Review messages

- Ensure that the RODM CMAS EYULOG shows EYUTS0030I and EYUTS0031I messages indicating the RODM interface is active. Issue a MONSPEC and ensure the MONSPEC points to the correct CMAS name for RODM. Issue a CPLEXDEF view and ensure that the CICSplex participates in RODM population of its resources. Issue a CMASD view for the RODM CMAS to ensure it contains the correct RODM name.
- If the RODM CMAS EYULOG indicates any of EYUTS0032I through EYUTS0036I messages, review the messages and take appropriate action.
- Check the NetView messages. NetView messages are issued to the destination defined by the NetView DEFAULTS or OVERRIDE command. The destination can be either the NetView network log, the MVS system log, or both. Verify that the messages EYUTR0001I and EYUTR0003I indicate successful initialization and CMAS contact. Look for any EYUTRxxxx messages indicating a failure and take appropriate action.
- Check the GMFHS log for any relative error messages and take appropriate action.

Collect trace output

- If none of the above steps resolve the problem, collect the following trace output and contact the IBM Support Center:
  - The EYUTRTC output file for the NetView job.
  - The AUXTRACE(s) output files for the RODM CMAS.

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.

```
26MAR1999  02:57:49 ----------- INFORMATION DISPLAY ---------------------------
COMMAND  ===>                                                 SCROLL ===>
CURR WIN ===> 1        ALT WIN ===>
 W1 =MAS===============EYUPLX01=EYUMAS1A=26MAR1999==02:57:49=CPSM====1=========
CMD Name     Type   CMAS     Status   MON RTA WLM Description
--- -------- ------ -------- -------- Act Act Act ----------------------
    EYUMAS1A LOCAL  EYUCMS1A ACTIVE   YES YES YES TOR1 - SYSA
```

Figure 81. MAS view showing active CICS system

   If the MAS view does not show an active status, see “Incomplete operations data returned” on page 133.

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 SM 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 82. Example of the Browse Analysis panel for the MAS view

   In the example shown in Figure 82, 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.

```
26MAR1999  16:00:54 ----------- INFORMATION DISPLAY ---------------------------
COMMAND  ===>                                                 SCROLL ===>
CURR WIN ===> 1        ALT WIN ===>
 W1 =ACTNDEF==========EYUPLX01=EYUPLX01=26MAR1999==16:00:54=CPSM====1=========
CMD Name      Event    View    Msg  Alert  ARM  Description
--- --------  -----  --------  ---  -----  ---  ------------------------ -------
    SOSPRI1   NO               YES  YES    NO   Critical SOS on production
```

Figure 83. Example of the ACTNDEF view

In the example shown in Figure 83, 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.

```
26MAR1999  04:53:08 ----------- INFORMATION DISPLAY ---------------------------
COMMAND  ===>                                                 SCROLL ===>
CURR WIN ===> 1        ALT WIN ===>
 W1 =RTAACTV==========EYUPLX01=EYUMAS1A=26MAR1999==04:53:08=CPSM====6=========
 CMD Name     System   Status    Period   Rate  Action   Def
 --- -------- -------- --------- -------- ----- -------- Type---
     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 84. 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.

```
26MAR1999  08:40:03 ----------- INFORMATION DISPLAY ---------------------------
COMMAND  ===>                                                 SCROLL ===>
CURR WIN ===> 1        ALT WIN ===>
 W1 =WLMSCOPE==========EYUPLX01=EYUPLX01=26MAR1999==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 85. 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 WLMAWORK view command.

```
26MAR1999  08:59:19 ----------- INFORMATION DISPLAY ---------------------------
COMMAND  ===>                                                 SCROLL ===>
CURR WIN ===> 1        ALT WIN ===>
 W1 =WLMAWORK==========EYUPLX01=EYUPLX01=26MAR1999==08:59:19=CPSM====1=========
CMD Name     Ownr Rout Targ Affinity Lifetime  Scope    Event    Status Algrthm
--- -------- ---- Cnt- Cnt- -------- --------- -------- Name---- ------ Type----
    EYUWLS02 HTC1    1    2 LUNAME   LOGON     EYUMAS2B          ACTIVE QUEUE
```

Figure 86. Example of the WLMAWORK 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 WLMAWORK view.

```
26MAR1999  09:03:20 ----------- INFORMATION DISPLAY ---------------------------
COMMAND  ===>                                                  SCROLL ===>
CURR WIN ===> 1        ALT WIN ===>
 W1 =WLMAWTOR==========EYUPLX01=EYUPLX01=26MAR1999==09:03:20=CPSM====1=========
CMD Workload Ownr Router   Connection
--- -------- ---- -------- Lost------
    EYUWLS02 HTC1 EYUMAS2B
```

Figure 87. 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.

```
26MAR1999  09:16:49 ----------- INFORMATION DISPLAY ---------------------------
COMMAND  ===>                                                  SCROLL ===>
CURR WIN ===> 1        ALT WIN ===>
 W1 =WLMATRAN==========EYUPLX01=EYUPLX01=26MAR1999==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 88. 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 WLMAWDEF command.

```
26MAR1999  11:42:55 ----------- INFORMATION DISPLAY ---------------------------
COMMAND  ===>                                                  SCROLL ===>
CURR WIN ===> 1        ALT WIN ===>
>W1 =WLMAWDEF==========EYUPLX01=EYUPLX01=26MAR1999==11:42:55=CPSM====4=========
CMD Name     Workload Ownr Trangrp  Luname            Userid   Target   Descrip
--- -------- -------- ---- -------- ----------------- -------- Scope--- -------
    T123DEF  EYUWLS02 HTC1          .++++T123         *        EYUMAS1B
    WMDFAFFA EYUWLS02 HTC1 WMTAFFA  .*                *        EYUMAS1B
    WMDFAFFB EYUWLS02 HTC1 WMTAFFB  .*                DEPT02*  EYUMAS2B
    WMDFAFFC EYUWLS02 HTC1 WMTAFFC  .*                *        EYUCSG02
```

Figure 89. Example of the WLMAWDEF 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 WLMAWDEF view shown in Figure 89.

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 WLMAWDEF 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 WLMAWORK 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 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/ESA 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 SM 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.

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 14. 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 90 on page 152 provides an overview of what will happen when you call the Support Center.

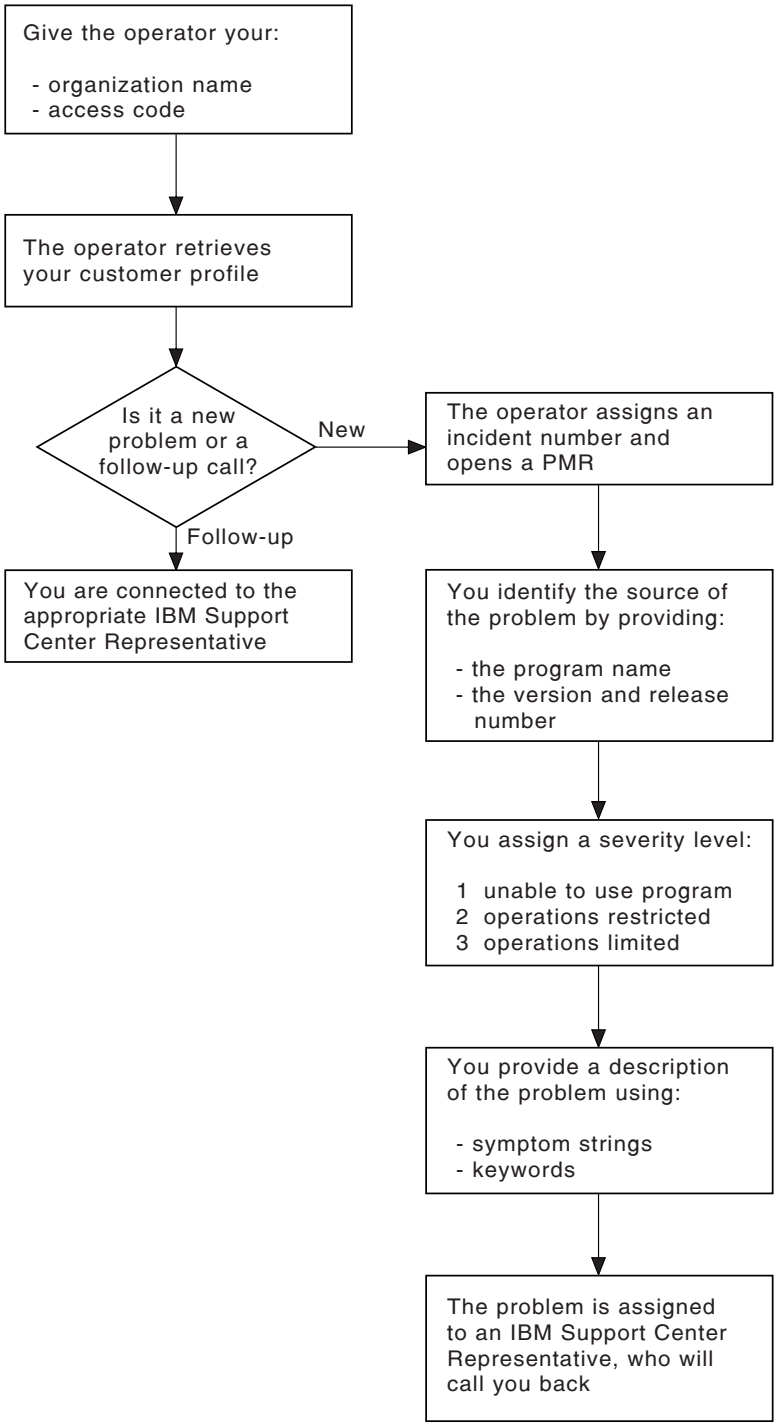

Figure 90. What happens when you call the Support Center

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
- 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 15, “APARs, fixes, and PTFs” on page 155.

Chapter 15. 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 156.
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 157.

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 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 OS/390: 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 OS/390: 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 162.

`xx` Is a unique identifier assigned by each component.

For example, EYU0MMIN is an executable module for the Monitor Services component.

Element type identifiers

ID	Description
$	Selection menus
0	Executable modules (C or assembler)
5	EUI record maps
6	Dynamically acquired control blocks or data areas
7	Module entry point descriptors
8	Function/service definition tables and assembled control blocks
9	Load modules
B or R	Assembler mapping DSECTs
C	C code generation macros
D	ISPF display or data entry panels
E	CLISTs
F	Function variables
G	ISPF message definitions
H	ISPF help panels
J	Screen definitions
M	C structure TYPEDEFs
P	Profile variables or USERFILE members
Q	Assembler code generation macros
S	EUI class tables

T	View, message, and action tables
U	Assembler equate files
V	C equate files
W or X	Assembled help modules
Z	View definitions

Component identifiers

CICSPlex SM component identifiers begin with one of three prefixes: BBC, BBM, or EYU.

BBC components

The BBC components are:

ID	Description
Q*x*	PlexManager Data Collectors
S*x*	Communications Server Controller
U*x*	End-user Interface Address Space Services
Z*x*	Global Services

BBM components

The BBM components are:

ID	Description
C*x*	Data Manager
H*x*	Information Services
L*x*	Linkage Services
M*x*	File Management
P*x*	Low-level Storage Management
Q*x*	PlexManager Selectors
S*x*	General Services
T*x*	TSO Support Functions
X*x*	Transaction Management
Z*x*	System and Application Control

EYU components

The EYU components are:

ID	Description
B*x*	Business Application Services
C*x*	Communications
E*x*	End-user Interface
M*x*	Monitor Services
N*x*	Managed Application System
P*x*	real-time analysis
T*x*	Topology Services
W*x*	Workload Manager
XC	Data Cache Manager
XD	Data Repository
XE	Environment Services System Services
XL	Kernel Linkage
XM	Message Services
XQ	Queue Manager
XS	Common Services
XZ	Trace Services

Major components of CICSPlex SM

The major components of CICSPlex SM and their 3-character identifiers are:

Component Name	Identifier
Business Application Services	BAS
Common Services	SRV
Communications	COM
Data Cache Manager	CHE
Data Repository	DAT
Environment Services System Services	ESS
Kernel Linkage	KNL
Managed Application System	MAS
Message Services	MSG
Monitor Services	MON
Queue Manager	QUE
real-time analysis	RTA
Topology Services	TOP
Trace Services	TRC

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.

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 the *CICSPlex SM Operations Views Reference* book.

The problem determination parameters

Table 5 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 5 (Page 1 of 2). 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
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	Take SDUMP on all CMAS and MAS failures	YES | NO	Both[1]
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[2]
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

Table 5 (Page 2 of 2). System parameters for problem determination

Name	Description	Values	Used by
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[3]
WLMCONMSG	Workload Manager console message flags	1–32	Both

Notes:

1. ESDUMP is valid only in local MASs. It is not valid in remote MASs.
2. RTATRACE is valid in a MAS only if status definitions are installed and being used by a user-written status program.
3. WLMTRACE is valid in a MAS only if it is a local MAS acting as a TOR in a CICSPlex SM workload.

Glossary

This glossary defines CICSPlex SM terms and abbreviations used in this book with other than their everyday meaning. Terms that are defined in the *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994, are not defined here unless CICSPlex SM usage is different from the meaning given there.

If you cannot find the definition you need, refer to the *Dictionary of Computing* or the *CICSPlex SM Master Index*, SC33-1812.

A

action command. A CICSPlex SM command that affects one or more of the resources represented in a view. Action commands can be issued from either the COMMAND field in the control area of the information display panel or the line command field in a displayed view. Valid action commands are listed with the description of each view. See also *overtype field*.

action definition (ACTNDEF). In real-time analysis, a definition of the type of external notification that is to be issued when the conditions identified in an analysis definition are true.

activity. See *BTS activity*.

adjacent CMAS. A CICSPlex SM address space (CMAS) that is connected to the local CMAS via a direct CMAS-to-CMAS link. Contrast with *indirect CMAS*. See also *local CMAS*.

alter expression. A character string that defines the changes to be made to a resource attribute. An alter expression is made up of one or more attribute expressions.

alternate window. A window to which the results of a hyperlink can be directed. By default, the results of a hyperlink are displayed in the same window from which the hyperlink is initiated. Contrast with *current window*.

alternate window (ALT WIN) field. In the control area of an information display panel, the field in which you can specify an alternate window to receive the results of a hyperlink.

analysis definition. In real-time analysis, a definition of the evaluations to be performed on specified CICS resources, the intervals at which those evaluations are to be performed, and the actions to be taken when a notifiable condition occurs.

analysis group. In real-time analysis, a group of one or more analysis definitions, status definitions, or both. Analysis definitions and status definitions must belong to an analysis group if they are to be installed automatically in a CICS system when that system starts.

analysis point monitoring (APM). In real-time analysis, resource monitoring across multiple CICS systems within a CICSplex that results in a single notification of a condition, rather than one notification for each system. Contrast with *MAS resource monitoring*.

analysis point specification. In real-time analysis, a specification that identifies the CMASs that are to be responsible for analysis point monitoring.

analysis specification. In real-time analysis, a specification that establishes system availability monitoring or MAS resource monitoring within a group of CICS systems.

AOR. Application-owning region.

API. Application programming interface

APM. Analysis point monitoring.

application-owning region (AOR). In a CICSplex configuration, a CICS region devoted to running applications. For dynamic routing, the terms *requesting region*, *routing region*, and *target region* are used instead of AOR to signify the role of the region in the dynamic routing request.

ARM. Automatic restart manager.

ASU. Automatic screen update.

attribute. See *resource attribute, resource table attribute*.

attribute expression. A reference to a resource table attribute and, in some cases, its value. Attribute expressions are used to build filter expressions, modification expressions, and order expressions.

attribute value. The data currently associated with a resource table attribute. For example, the file attribute OPENSTATUS might have a value of CLOSED.

automatic restart manager (ARM). A recovery function of MVS/ESA 5.2 that provides improved availability for batch jobs and started tasks by restarting them automatically if they end unexpectedly. The affected batch job or started task can be restarted on the same system or on a different one, if the system itself has failed.

automatic screen update (ASU). A CICSPlex SM facility that automatically updates the data in all unlocked windows at user-defined intervals. See also *automatic screen update interval*.

automatic screen update interval. The time interval between one automatic screen update and the next. This interval can be set in the CICSPlex SM user profile or when the ASU facility is turned on. See also *automatic screen update (ASU)*.

B

BAS. Business Application Services

batched repository-update facility. A CICSPlex SM facility, invoked from the CICSPlex SM end user interface, for the bulk application of CICSPlex SM definitions to a CMAS data repository.

BTS. CICS business transaction services

BTS activity. One part of a process managed by CICS BTS. Typically, an activity is part of a *business transaction*.

BTS process. A collection of more than one CICS BTS *activities*. Typically, a process is an instance of a *business transaction*.

BTS set. See CICS system group

business application. Any set of CICS resources that represent a meaningful entity to an enterprise or a user (such as, Payroll).

Business Application Services (BAS). The component of CICSPlex SM that provides the ability to define and manage business applications in terms of their CICS resources and associated CICS systems. BAS provides a central definition repository for CICS systems, complete with installation facilities and the ability to restrict a CICSPlex SM request to those resources defined as being part of the business application. See also *business application, scope*.

business transaction. A self-contained business function, for example, the booking of an airline ticket.

C

CAS. Coordinating address space.

CBIPO. Custom-built installation process offering.

CBPDO. Custom-built product delivery offering.

CEDA. A CICS transaction that defines resources online. Using CEDA, you can update both the CICS system definition data set (CSD) and the running CICS system.

CICS Business Transaction Services (BTS). A CICS domain that supports an application programming interface (API) and services that simplify the development of *business transactions*.

CICS system. The entire collection of hardware and software required by CICS. In CICSPlex SM topology, a definition referring to a CICS system that is to be managed by CICSPlex SM. See also *CICSplex, CICS system group*.

CICS system group. A set of CICS systems within a CICSplex that can be managed as a single entity. In CICSPlex SM topology, the user-defined name, description, and content information for a CICS system group. A CICS system group can be made up of CICS systems or other CICS system groups. In CICS business transaction services (BTS), a BTS set, that is the set of CICS regions across which BTS processes and activities may execute. See also *CICSplex, CICS system*.

CICSplex. A CICS complex. A CICSplex consists of two or more CICS regions that are linked using CICS intercommunication facilities. The links can be either intersystem communication (ISC) or interregion communication (IRC) links, but within a CICSplex are more commonly IRC. Typically, a CICSplex has at least one terminal-owning region (TOR), more than one application-owning region (AOR), and may have one or more regions that own the resources being accessed by the AORs. In CICSPlex SM, a management domain. The largest set of CICS regions, or CICS systems, to be manipulated by CICSPlex SM as a single entity. CICS systems in a CICSplex being managed by CICSPlex SM do not need to be connected to each other. See also *CICS system, CICS system group*.

CICSPlex SM. IBM CICSPlex System Manager.

CICSPlex SM address space (CMAS). A CICSPlex SM component that is responsible for managing CICSplexes. A CMAS provides the single-system image for a CICSplex by serving as the interface to other CICSplexes and external programs. There must be at least one CMAS in each MVS image on which you are running CICSPlex SM. A single CMAS can manage CICS systems within one or more CICSplexes. See also *coordinating address space (CAS), managed application system (MAS)*.

CICSPlex SM token. Unique, 4-byte values that CICSPlex SM assigns to various elements in the API environment. Token values are used by CICSPlex SM to correlate the results of certain API operations with subsequent requests.

client program. In dynamic routing, the application program, running in the *requesting region*, that issues a remote link request.

CMAS. CICSPlex SM address space.

CMAS link. A communications link between one CICSPlex SM address space (CMAS) and another CMAS or a remote managed application system (remote MAS). CMAS links are defined when CICSPlex SM is configured.

CODB. A CICSPlex SM transaction for interactive, system-level debugging of CMASs and of CICS/ESA, CICS/MVS, and CICS/VSE MASs. CODB must be used only at the request of customer support personnel.

COD0. A CICSPlex SM transaction for interactive, method-level debugging of CMASs and of CICS/ESA, CICS/MVS, CICS/VSE, and CICS for OS/2 MASs. COD0 must be used only at the request of customer support personnel.

COLU. A CICSPlex SM transaction for generating reports about CMAS and local MAS components. COLU must be used only at the request of customer support personnel.

COMMAND field. In the control area of an information display panel, the field that accepts CICSPlex SM, ISPF, and TSO commands. Contrast with *option field*.

command-level interface. A CICSPlex SM API interface that uses the CICS translator to translate EXEC CPSM statements into an appropriate sequence of instructions in the source language.

Common Services. A component of CICSPlex SM that provides commonly requested services (such as GETMAIN, FREEMAIN, POST, and WAIT processing) to other CICSPlex SM components.

communication area (COMMAREA). A CICS area that is used to pass data between tasks that communicate with a given terminal. The area can also be used to pass data between programs within a task.

Communications. A component of CICSPlex SM that provides all services for implementing CMAS-to-CMAS and CMAS-to-MAS communication.

context. A named part of the CICSPlex SM environment that is currently being acted upon by CICSPlex SM. For configuration tasks, the context is a CICSPlex SM address space (CMAS); for all other tasks, it is a CICSplex. See also *scope*.

control area. The top three lines of an information display panel, containing the panel title, the screen update time, the short message area, the COMMAND and SCROLL fields, and the current window (CUR WIN) and alternate window (ALT WIN) fields.

coordinating address space (CAS). An MVS subsystem that provides ISPF end-user access to the CICSplex to be accessed. See also *CICSPlex SM address space, managed application system (MAS)*.

coordinating address space subsystem ID. Identifies the coordinating address space (CAS) which can be up to 4 characters, to be connected to when issuing CICSPlex SM requests. The name of the CAS is installation-dependent, and is defined in the CICSPlex SM user profile.

cross-system coupling facility (XCF). XCF is a component of MVS that provides functions to support cooperation between authorized programs running within a sysplex.

current window. The window to which the results of all commands issued in the COMMAND field are directed, unless otherwise requested. Contrast with *alternate window*.

current window (CUR WIN) field. In the control area of an information display panel, the field that contains the window number of the current window. You can change the number in this field to establish a new current window.

custom-built installation process offering (CBIPO). A product that simplifies the ordering, installation, and service of MVS system control programs and licensed programs by providing them with current updates and corrections to the software that is already integrated.

custom-built product delivery offering (CBPDO). A customized package of both products and service, or of service only, for MVS system control programs and licensed programs.

D

Data Cache Manager. A component of CICSPlex SM that manages logical cache storage for use by other CICSPlex SM components.

data repository. In CICSPlex SM, the VSAM data set that stores administrative data, such as topology and monitor definitions, for a CICSPlex SM address space (CMAS).

Data Repository. A component of CICSPlex SM that provides methods for creating, accessing, updating, and deleting data in the CICSPlex SM data repository. See also *Managed Object Services*.

Database Control (DBCTL). An IMS/ESA facility providing an interface between CICS/ESA and IMS/ESA

that allows access to IMS DL/I full-function databases and to data-entry databases (DEDBs) from one or more CICS/ESA systems.

Database 2 (DB2). An IBM licensed program. DB2 is a full-function relational database management system that presents a data structure as a table consisting of a number of rows (or records) and a number of columns.

DBCTL. Database Control.

DB2. Database 2.

derived field. On a monitor view, a field whose value does not come directly from CICS or CICSPlex SM data, but is calculated based on the values in other fields. See also *derived value*.

derived value. A rate, average, or percentage that results from CICSPlex SM processing of CICS statistics.

display area. On an information display panel, the area where windows can be opened to display data. The display area appears below the control area. The bottom two lines of the display area can be used to display the PF key assignments in effect for a CICSPlex SM session.

display attributes. A CICSPlex SM user profile option that controls the appearance of the window information line, field headings, and threshold values in a view.

display command. A CICSPlex SM command that extends the ISPF interface to create and control a multiwindow environment.

distributed program link (DPL). Function of CICS intersystem communication that enables CICS to ship LINK requests between CICS regions.

distributed routing program (DSRTPGM). A CICS-supplied user-replaceable program that can be used to dynamically route:

- CICS BTS processes and activities
- Transactions started by non-terminal related EXEC CICS START commands

DPL. Distributed program link.

DTR. Dynamic transaction routing.

dynamic routing. The automatic routing of a transaction or program, at the time it is initiated, from a requesting region to a suitable target region. Routing terminal data to an alternative transaction at the time the transaction is invoked. To do this, CICS allows the dynamic routing program to intercept the terminal data and redirect it to any system and transaction it chooses. See also dynamic routing program (EYU9XLOP)

dynamic routing program (EYU9XLOP). A user-replaceable CICS program that selects dynamically both the system to which a routing request is to be sent and the transaction's remote name. The alternative to using this program is to make these selections when a remote transaction is defined to CICS (static routing). See also *static routing*

dynamic transaction routing (DTR). The automatic routing of a transaction, at the time it is initiated, from a transaction-owning region (TOR) to a suitable application-owning region (AOR).

E

Environment Services System Services (ESSS). A component of CICSPlex SM that implements the formal MVS/ESA subsystem functions required by the product. ESSS provides cross-memory services, data space management, connection services, and lock management. An ESSS system address space is created at CICSPlex SM initialization and remains in the MVS image for the life of the IPL.

ESSS. Environment Services System Services.

evaluation definition. In real-time analysis, a definition of the resources that are to be sampled. When the result of an evaluation is true, an associated analysis definition is used to determine whether a notifiable condition has occurred.

event. A significant occurrence within the CICSplex or system for which the user has requested notification. For example, the end of processing, a subsystem failure, or any unusual condition in the system could be defined by a user as an event.

event notification. A CICSPlex SM notification of a significant occurrence within a CICSplex or CICS system.

extended diagnostic mode (XDM). A CICSPlex SM online internal diagnostic facility. XDM provides no information about resources managed by CICSPlex SM, and should be turned on only at the request of IBM customer support personnel. XDM can be turned on and off in the CICSPlex SM user profile.

external notification. In RTA, an event notification, generic alert, or operator message issued when a notifiable condition occurs.

F

file-owning region. In a CICSplex configuration, a CICS system devoted to managing CICS file access.

filter expression. A character string that consists of logical expressions to be used in filtering resource table records. A filter expression is made up of one or more attribute expressions.

FOR. File-owning region.

form. The way in which data obtained from a query is presented in a view. See also *query, view*.

G

generic alert. A Systems Network Architecture (SNA) Network Management Vector that enables a product to signal a problem to the network. CICSPlex SM uses generic alerts as part of its interface to NetView.

GMFHS. Graphic Monitor Facility host subsystem.

goal algorithm. In CICSPlex SM's workload balancing, an algorithm used to select an AOR to process a dynamic transaction. Using the goal algorithm, CICSPlex SM selects the AOR that is the least affected by conditions such as short-on-storage, SYSDUMP, and TRANDUMP; is the least likely to cause the transaction to abend; and is most likely to enable the transaction to meet response-time goals set for it using the Workload Manager component of MVS/ESA SP 5.1. Contrast with *queue algorithm*.

Graphic Monitor Facility host subsystem. A NetView feature that manages configuration and status updates for non-SNA resources.

H

hyperlink. A direct connection between the data in one CICSPlex SM view and a view containing related information. For example, from a view that lists multiple CICS resources, there may be a hyperlink to a detailed view for one of the resources. To use a hyperlink, place the cursor in the data portion of a hyperlink field and press Enter.

hyperlink field. On a CICSPlex SM view, a field for which a hyperlink is defined. The headings of hyperlink fields are shown in high intensity or color, depending on the terminal type.

I

IBM CICSPlex System Manager for MVS/ESA (CICSPlex SM). An IBM CICS system-management product that provides a single-system image and a single point of control for one or more CICSplexes that can be installed on heterogeneous operating systems.

indirect CMAS. A CICSPlex SM address space (CMAS) that the local CMAS can communicate with via an adjacent CMAS. There is no direct CMAS-to-CMAS link between the local CMAS and an indirect CMAS. Contrast with *adjacent CMAS*. See also *local CMAS*.

information display panel. The panel that supports the CICSPlex SM window environment. It consists of a control area and a display area. CICSPlex SM views are displayed in windows within the display area of this panel.

information display parameters. A CICSPlex SM user profile option that defines the initial screen configuration, how frequently the screen will be updated by ASU, and how long a window will wait for command processing to complete before timing out.

installation verification procedure (IVP). A procedure distributed with a system that tests the newly generated system to verify that the basic facilities of the system are functioning correctly.

interregion communication. Synonym for *multiregion operation*.

intersystem communication (ISC). Communication between separate systems by means of SNA networking facilities or by means of the application-to-application facilities of an SNA access method.

intertransaction affinity. A relationship between CICS transactions, usually the result of the ways in which information is passed between those transactions, that requires them to execute in the same CICS region. Intertransaction affinity imposes restrictions on the dynamic routing of transactions.

IRC. Interregion communication.

ISC. Intersystem communication.

IVP. Installation verification procedure.

K

Kernel Linkage. A component of CICSPlex SM that is responsible for building data structures and managing the interfaces between the other CICSPlex SM components. The environment built by Kernel Linkage is known as the method call environment.

L

line command field. In a CICSPlex SM view, the 3 character field, to the left of the data, that accepts action commands.

local CMAS. The CICSPlex SM address space (CMAS) that a user identifies as the current context when performing CMAS configuration tasks.

local MAS. A managed application system (MAS) that resides in the same MVS image as the CICSPlex SM address space (CMAS) that controls it and that uses the Environment Services System Services (ESSS) to communicate with the CMAS.

logical scope. A set of logically related CICS resources that are identified in a CICSPlex SM resource description. A logical scope can be used to qualify the context of a CICSPlex SM request.

M

maintenance point. A CICSPlex SM address space (CMAS) that is responsible for maintaining CICSPlex SM definitions in its data repository and distributing them to other CMASs involved in the management of a CICSplex. See also *data repository*.

Major object descriptor block (MODB). In CICSPlex SM, a control structure built by Kernel Linkage during initialization of a CICSPlex SM component that contains a directory of all methods that make up that component. The structure of the MODB is the same for all components.

Major object environment block (MOEB). In CICSPlex SM, a control structure built by Kernel Linkage during initialization of a CICSPlex SM component and pointed to by the MODB. The MOEB stores information critical to a CICSPlex SM component and anchors data used by the component. The structure of the MOEB is unique to the component it supports.

MAL. Message argument list.

managed application system (MAS). A CICS system that is being managed by CICSPlex SM. See *local MAS, remote MAS*.

managed object. A CICSPlex SM-managed CICS resource or a CICSPlex SM definition represented by a resource table. A view is based on a single managed object.

Managed Object Services. A subcomponent of the Data Repository component of CICSPlex SM that translates a request for data (from real-time analysis, for example) into the method calls required to obtain the data.

MAS. Managed application system.

MAS agent. A CICSPlex SM component that acts within a CICS system to provide monitoring and data collection for the CICSPlex SM address space (CMAS). The level of service provided by a MAS agent depends on the level of CICS the system is running under and whether it is a local or remote MAS. See also *CICSPlex SM address space (CMAS), local MAS, remote MAS*.

MAS resource monitoring (MRM). In real-time analysis, resource monitoring at the CICS system level; it results in one notification of a condition for each system in which it occurs. If the same condition occurs in three CICS systems where MAS resource monitoring is active, three notifications are issued. Contrast with *analysis point monitoring*.

Message argument list (MAL). In CICSPlex SM, a data structure passed between methods using Kernel Linkage method call services.

message line. On an information display panel, the line in the control area where a long message appears when the HELP command is issued in response to a short message. The message line temporarily overlays the CURR WIN and ALT WIN fields.

Message Services. A component of CICSPlex SM that provides services for building and issuing MVS/ESA console messages to other CICSPlex SM components.

meta-data. Internal data that describes the structure and characteristics of CICSPlex SM managed objects.

method. (Action.) An application programming interface (API) instruction that resolves into an EXEC CICS command, issued against one or more resources in one or more CICS systems, within the current context and scope.

method. In CICSPlex SM, one of the programs that make up a CICSPlex SM component. See also *message argument list (MAL)*.

mirror transaction. CICS transaction that recreates a request that is function shipped from one system to another, issues the request on the second system, and passes the acquired data back to the first system.

MODB. Major object descriptor block.

modification expression. A character string that defines the changes to be made to a resource attribute. A modification expression is made up of one or more attribute expressions.

MOEB. Major object environment block.

monitor definition. A user-defined statement of the specific resource occurrences (such as the program named PAYROLL) to be monitored by CICSPlex SM. A monitor definition can either be linked to a monitor specification as part of a monitor group or be installed directly into an active CICS system. See also *monitor group, monitor specification*.

monitor group. A user-defined set of CICSPlex SM monitor definitions that can either be linked to a monitor specification for automatic installation or be installed directly into an active CICS system. See also *monitor definition, monitor specification*.

monitor interval. The number of minutes that are to elapse before the statistics counters containing accumulated resource monitoring data are automatically reset. This value is part of a CICSplex definition and affects all of the CICS systems and CICS system groups associated with that CICSplex. See also *period definition, sample interval*.

monitor specification. A user-defined statement of the types of resources (such as programs) to be monitored by CICSPlex SM and how often data should be collected. A monitor specification is associated with a CICS system and is automatically installed each time the CICS system starts up. See also *monitor definition, monitor group*.

Monitoring Services. A component of CICSPlex SM that is responsible for monitoring resources within a CICS system and making the collected data available to other CICSPlex SM components.

MRM. MAS resource monitoring.

MRO. Multiregion operation.

MSM. MultiSystem Manager.

multiregion operation (MRO). Communication between CICS systems without the use of SNA network facilities. Synonymous with *interregion communication*.

MultiSystem Manager. An object-oriented, graphical systems management application that runs under NetView for MVS.

MVS image. A single instance of the MVS operating system.

MVS system. An MVS image together with its associated hardware.

N

NetView. An IBM network management product that can provide rapid notification of events and automated operations. CICSPlex SM can be set up to send generic alerts to NetView as part of its event processing capabilities.

NetView Graphic Monitor Facility (NGMF). A function of the NetView program that provides the network operator with a graphic topological presentation of a network controlled by the NetView program and that allows the operator to manage the network interactively.

NetView program. An IBM licensed program used to monitor and manage a network and to diagnose network problems.

NGMF. NetView Graphic Monitor Facility.

notification. A message that is generated asynchronously by a CICSPlex SM managed object to describe an event related to the object.

O

option field. On a CICSPlex SM menu, the field in which you can specify an option number or letter. Contrast with *command field*.

order expression. A character string that defines either the attributes to be used in sorting resource table records, or the attributes to be included in a resource table view. An order expression is made up of one or more attribute expressions.

override expression. A character string that defines the changes to be made to a resource attribute. An override expression is made up of one or more attribute expressions.

overtype field. On a CICSPlex SM view, a field containing a value that can be changed by typing a new value directly into the field. Values that can be overtyped are shown in high intensity or color, depending on the terminal type. Acceptable values for overtype fields are listed with the description of each view. See also *action command*.

P

parameter expression. A character string that defines the parameters required for an action to complete or a definition to be processed.

parameter repository. In CICSPlex SM, a data set that stores cross-system communication definitions that allow one coordinating address space (CAS) to communicate with other CASs.

period definition. A user-defined range of hours and minutes and the time zone to which that range applies. A period definition is used to indicate when an action, such as resource monitoring, is to occur. See also *monitor interval, sample interval*.

PlexManager. A service utility that can be used to manage the communication connections between multiple coordinating address spaces (CASs) and between a CAS and its associated CICSPlex SM address spaces (CMASs) and CICSplexes.

process. See *BTS process*

processing thread. A connection between an application program and the CICSPlex SM API. A program can establish multiple processing threads, but each one is considered a unique API user; no resources can be shared across the boundary of a thread.

pseudoconversation. A CICS application designed to appear to the user as a continuous conversation, but that consists internally of multiple separate tasks.

Q

query. A request for specific data that is generated by a view command. See also *form, view*.

queue algorithm. In CICSPlex SM's workload balancing, an algorithm used to select an AOR to process a dynamic transaction. Using the queue algorithm, CICSPlex SM selects the AOR that has the shortest queue of transactions (normalized to MAXTASKs) waiting to be processed; is the least affected by conditions such as short-on-storage, SYSDUMP, and TRANDUMP; and is the least likely to cause the transaction to abend. Contrast with *goal algorithm*.

Queue Manager. A component of CICSPlex SM that creates and manages queues of data in a cache that is shared by a CMAS and its local MASs.

R

RACF. Resource Access Control Facility.

real-time analysis (RTA). A component of CICSPlex SM that is responsible for monitoring the status of a CICS system or resource against its desired status, and issuing one or more external notifications when deviations occur.

record pointer. An internal indicator of the next resource table record to be processed in a result set.

related scope. A CICS system where resources defined to CICSPlex SM as remote should be assigned and, optionally, installed as local resources. See also *target scope*.

remote MAS. A managed application system (MAS) that uses MRO or LU 6.2 to communicate with the CICSPlex SM address space (CMAS) that controls it. A remote MAS may or may not reside in the same MVS image as the CMAS that controls it.

requesting region. The region in which a dynamic routing request originates. For dynamic transaction routing and inbound client dynamic program link requests, this is typically a TOR; for dynamic START requests and peer-to-peer dynamic program link requests, this is typically an AOR.

resource. Any physical or logical item in a CICS system, such as a transient data queue, a buffer pool, a file, a program, or a transaction.

Resource Access Control Facility (RACF). An IBM licensed program that provides for access control by identifying and verifying the users to the system, authorizing access to protected resources, logging any detected unauthorized attempts to enter the system, and logging the detected accesses to protected resources.

resource assignment. A user-defined statement that selects resource definitions to be assigned to CICS systems and, optionally, specifies resource attributes to override those definitions. A resource assignment applies to a single resource type and must be associated with a resource description. See also *resource definition, resource description*.

resource attribute. A characteristic of a CICS resource, such as the size of a buffer pool.

resource definition. In CICSPlex SM, a user-defined statement of the physical and operational characteristics of a CICS resource. Resource definitions can be associated with resource descriptions as part of a resource group. See also *resource description, resource group*.

resource description. A user-defined set of CICSPlex SM resource definitions that can be automatically installed in CICS systems and named as a logical scope for CICSPlex SM requests. Resource descriptions represent the largest set of CICS resources that can be managed by CICSPlex SM as a single entity. A resource description can be associated with one or more resource assignments. See also *logical scope, resource assignment, resource definition*.

resource group. A user-defined set of CICSPlex SM resource definitions. A resource group can be associated with resource descriptions either directly or by means of resource assignments. See also *resource assignment, resource definition, resource description*.

Resource Object Data Manager (RODM). A component of the NetView program that operates as a cache manager and that supports automation applications. RODM provides an in-memory cache for maintaining real-time data in an address space that is accessible by multiple applications.

resource table. The external representation of a CICSPlex SM managed object. A resource table defines all the attributes, or characteristics, of a managed object.

resource table attribute. A characteristic of a CICSPlex SM managed object, as represented by a field in a resource table.

resource type. A group of related resources, such as files.

result set. A logical group of resource table records that can be accessed, reviewed, and manipulated by an API program.

retention period. For a monitored CICS system, the period of time for which monitor data is retained after the system becomes inactive. If a system is being monitored, becomes inactive, and remains inactive beyond the specified retention period, the monitor data is discarded. If the system becomes active before the retention period expires, the monitor data gathered before the system became inactive is retained, and monitoring continues.

RODM. Resource Object Data Manager.

routing region. The region in which the decision is made as to which is the most suitable target region for a dynamic routing request. For dynamic transaction routing, dynamic START requests, and inbound client dynamic program link requests, this is typcially a TOR; for dynamic peer-to-peer program link requests, this is typically an AOR.

RTA. real-time analysis.

run-time Interface. A CICSPlex SM API interface that accepts commands in the form of text strings and generates the appropriate API calls. The run-time interface supports programs written as REXX EXECs.

S

SAM. System availability monitoring.

sample interval. The duration, in seconds, between occurrences of data collection for a specific resource type. See also *monitor interval, period definition, resource type*.

scope. A named part of the CICSPlex SM environment that qualifies the context of a CICSPlex SM request. The scope can be the CICSplex itself, a CICS system, a CICS system group, or any set of CICS resources that are defined as a logical scope in a CICSPlex SM resource description. For configuration tasks, where the context is a CICSPlex SM address space (CMAS), the scope is ignored. When you are applying security, scope must be a single CICS system or CICSplex. It cannot be a CICS system group or any combination of individual CICSplexes or CICS systems. See also *context, logical scope*.

screen configuration. A user-defined, named layout of windows and the context, scope, view, and sort order associated with each. The initial configuration to be displayed when CICSPlex SM is accessed can be identified on the user profile.

screen repository. In CICSPlex SM, a data set that stores screen configuration definitions created by the SAVESCR display command. See also *screen configuration*.

selection list. In CICSPlex SM, a data set that stores cross-system communication definitions that allow one coordinating address space (CAS) to communicate with other CASs.

selection list. A list of named items, such as views or screen configurations, from which one can be selected.

server program. In dynamic routing, the application program specified on the link request, and which is executed in the *target region*.

service point. One of the combinations of products and contexts that is known to the coordinating address space (CAS) to which you are connected. See also *context*.

session control parameters. A CICSPlex SM user profile option that sets the coordinating address space (CAS) subsystem ID used for accessing CICSPlex SM views and controls the extended diagnostic mode (XDM).

short message area. In the control area of an information display panel, that part of the title line that displays short messages.

single point of control. The ability to access and manage all CICS systems and their resources in a CICSplex from a single terminal or user session.

single system image. The collection and presentation of data about multiple CICS systems as though they were a single CICS system. In CICSPlex SM, the single-system image is provided by the CICSPlex SM address space (CMAS).

specification. See *analysis specification, monitor specification, workload specification*.

Starter Set. A part of CICSPlex SM comprising sample CICSPlex SM definitions and sample JCL. The Starter Set samples may be used as supplied for educational purposes. They may also be copied and adapted for the customer environment.

static routing. Non-dynamic routing. The routing request is routed to a predetermined system. Static transaction routing occurs when NO is specified is the Dynamic field in either the transaction definition or the progam definition. In both cases, the request is routed to the system named in the Remote Sysid field.

status definition. In real-time analysis, a definition of a user-written program to be invoked at specified intervals to evaluate the status of a non-CICS resource.

summarized result set. A special type of result set that is produced by grouping, or summarizing, the resource table records in a result set. See also *result set*.

summary expression. A character string that consists of one or more summary options and the resource table attributes to which they apply. See also *summary option*.

summary option. A value that indicates how the attribute values in a resource table are to be summarized.

sysplex. A set of MVS systems communicating and cooperating with each other through specific multisystem hardware components and software services to process customer workloads.

system availability monitoring (SAM). In real-time analysis, the monitoring of CICS systems to determine whether: they are active during their defined hours of operation; they are experiencing a short-on-storage, SYSDUMP, TRANDUMP, MAXTASK, or STALL condition. If a CICS system becomes inactive or one of the specified conditions occurs, an external notification is issued.

system image. The representation of a program and its related data as it exists in main storage.

T

target region. The region selected from a set of target regions as the most suitable region in which to execute the work request. For all dynamic routing requests, this is typically an AOR.

target scope. A CICS system or CICS system group where resources defined to CICSPlex SM should be assigned and, optionally, installed. See also *related scope*.

temporary maintenance point. A CICSPlex SM address space (CMAS) that serves as the maintenance point when the identified maintenance point is unavailable. See also *maintenance point*.

terminal-owning region. In a CICSplex configuration, a CICS region devoted to managing the terminal network. For dynamic routing, the terms *requesting region* and *routing region* are used instead of TOR to signify the role of the region in the dynamic routing request.

thread. See *processing thread*.

time-period definition. A user-defined range of hours and minutes, and the time zone to which that range applies. A time-period definition is used to indicate when an action, such as resource monitoring, is to occur.

token. See *CICSPlex SM token, user token*.

topology. An inventory of CICS and CICSPlex SM resources, and a map of their relationships. CICSPlex SM supports the definition of resource and system topology.

topology definition. A named subset of CICS and CICSPlex SM resources. Topology definitions are user-created and can include CICSplexes, CICS systems, and CICS system groups.

Topology Services. A component of CICSPlex SM that is responsible for maintaining topology information about CICSplexes and resources, and making it available to other CICSPlex SM components.

TOR. Terminal-owning region.

Trace Services. A component of CICSPlex SM that provides other CICSPlex SM components with the ability to write trace records to the CICS trace table and trace data sets. Trace Services also writes trace

records created by a MAS to the trace table and data set of the managing CMAS.

transaction group. A user-defined, named set of transactions that determines the scope of workload balancing and the affinity relationships between transactions.

U

user token. Unique, 1- to 4-byte values that an API user can assign to asynchronous requests. User token values are not used by CICSPlex SM; they are simply held until the request is complete and then returned to the user.

V

view. In the CICSPlex SM API, a temporary, customized form of a resource table. A view can consist of some or all of the resource table attributes in any order. In the CICSPlex SM ISPF end-user interface, a formatted display of selected data about CICS resources or CICSPlex SM definitions. The data in a view is obtained from a query and can be presented in one or more forms. The data can be limited to a subset of CICSplex resources or definitions by establishing a context and scope.

view command. A CICSPlex SM command that displays a view in a window of the display area. The name of the view displayed matches the name of the view command. See also *view*.

W

window. In CICSPlex SM, a subdivision of the display area. The results of any CICSPlex SM view or display command are directed to a single window, which is the current window by default. Contrast with *view*. See also *current window*, *alternate window*.

window identifier. On a window information line, the field that identifies the window. A window identifier consists of a one-character status code and a number in the range 1 through 20.

window information line. The top line of each window in the display area. It includes the window identifier, the name of the view displayed in the window, the context and scope in effect, the date and time when the view was last refreshed, and the product name.

window number. A number assigned by CICSPlex SM to a window when it is opened. The window number is the second part of the window identifier on the window information line.

window status code. A one-character code that indicates whether a window is ready to receive commands, is busy processing commands, is not to be updated, or contains no data. It also indicates when an error has occurred in a window. The window status code is the first character of the window identifier on the window information line.

WLM. Workload Manager.

workload. The total number of transactions that a given CICSplex is intended to process in a specific period. For example, a workload could be expressed as a number of transactions per hour, or per day. In CICSPlex SM, a named set of transactions and CICS systems, acting as requesting regions, routing regions, and target regions that form a single, dynamic entity.

workload balancing. The technique of balancing a workload across multiple target regions that are capable of processing the work.

workload definition. A user-defined statement of the transaction groups associated with a CICS system that is an AOR. A workload definition can either be linked to a workload specification as part of a workload group or be installed directly into an active workload. See also *workload group, workload specification*.

workload group. A user-defined set of CICSPlex SM workload definitions that can either be linked to a workload specification for automatic installation or be installed directly into an active workload. See also *workload definition, workload specification*.

Workload Manager (WLM). A component of CICSPlex SM that is responsible for managing the transaction workload in a CICSplex through the use of dynamic transaction routing.

workload separation. The technique of separating a workload into discrete parts, and allocating specific transactions to specific AORs.

workload specification. A user-defined statement that identifies a workload and a set of CICS systems acting as AORs. A workload specification also provides default management criteria for transactions that are not defined to CICSPlex SM. It is associated with a CICS system that is a TOR and is automatically installed each time the CICS system starts up. See also *workload definition, workload group*.

X

XCF. Cross-system coupling facility of MVS/ESA.

XDM. Extended diagnostic mode

Index

E

F

H

I

K

L

M

N

O

U

V

W

X

Sending your comments to IBM

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book.

Please limit your comments to the information in this book and the way in which the information is presented.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

You can send your comments to IBM in any of the following ways:

- By mail, to this address:

  Information Development Department (MP095)
  IBM United Kingdom Laboratories
  Hursley Park
  WINCHESTER,
  Hampshire
  SO21 2JN
  United Kingdom

- By fax:
  - From outside the U.K., after your international access code use 44–1962–870229
  - From within the U.K., use 01962–870229
- Electronically, use the appropriate network ID:
  - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
  - IBMLink™: HURSLEY(IDRCF)
  - Internet: idrcf@hursley.ibm.com

Whichever you use, ensure that you include:

- The publication number and title
- The topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.

Program Number: 5655-147

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC33-0791-03

Spine information:

CICS TS for OS/390 CICSPlex® SM Problem Determination *Release 3*