



Tivoli Business System Manager
CICSplex SM Release Notes Version 1.5



Tivoli Business System Manager
CICSplex SM Release Notes Version 1.5

CICSplex SM Release Notes Version 1.1

Copyright Notice

© Copyright IBM Corporation 2001. All rights reserved. May only be used pursuant to a Tivoli Systems Software License Agreement, an IBM Software License Agreement, or Addendum for Tivoli Products to IBM Customer or License Agreement. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without prior written permission of IBM Corporation. IBM Corporation grants you limited permission to make hardcopy or other reproductions of any machine-readable documentation for your own use, provided that each such reproduction shall carry the IBM Corporation copyright notice. No other rights under copyright are granted without prior written permission of IBM Corporation. The document is not intended for production and is furnished "as is" without warranty of any kind. **All warranties on this document are hereby disclaimed, including the warranties of merchantability and fitness for a particular purpose.**

U.S. Government Users Restricted Rights—Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corporation.

Trademarks

IBM, the IBM logo, Tivoli, the Tivoli logo, AIX, Cross-Site, NetView, and TME are trademarks or registered trademarks of International Business Machines Corporation or Tivoli Systems Inc. in the United States, other countries, or both.

Lotus is a registered trademark of Lotus Development Corporation.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

Notices

References in this publication to Tivoli Systems or IBM products, programs, or services do not imply that they will be available in all countries in which Tivoli Systems or IBM operates. Any reference to these products, programs, or services is not intended to imply that only Tivoli Systems or IBM products, programs, or services can be used. Subject to valid intellectual property or other legally protectable right of Tivoli Systems or IBM, any functionally equivalent product, program, or service can be used instead of the referenced product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by Tivoli Systems or IBM, are the responsibility of the user. Tivoli Systems or IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, New York 10504-1785, U.S.A.

© Copyright International Business Machines Corporation 2001. All rights reserved.

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

Contents

Chapter 1. Overview of CICSplex SM.....	1
Chapter 2. Overview of TBSM Processing	3
Object Discovery.....	3
Object Rediscovery.....	3
Processing Events.....	3
Event Flow.....	4
Command Execution.....	5
Chapter 3. Installation	7
SMP/E Install.....	7
Customizing Tivoli NetView for OS/390 Components.....	7
Installing the REXX Function Package.....	8
SMP/E Control Statements.....	8
Alias Considerations.....	9
Host Command Entry.....	9
Chapter 4. OS/390 Customization	11
Setting up the Discovery Job.....	11
Using the SubName Capability.....	12
Defining the SubName.....	12
Creating the Discovery Job.....	13
Job Statement.....	13
Extract Step.....	13
STEPLIB DD Statement.....	13
SYSPRINT DD Statement.....	14
AOPDFILE DD Statement.....	14
Transmit Step.....	14
GTMAOPE0 CONTROL CARDS.....	14
Discovery Job Example.....	16
Discovery Job Sample Output.....	17
GTMAOPE0 Job Sample Output.....	17
CICSplex SM Definition for TBSM.....	17
CICSplex SM Objects Required for TBSM.....	18
Defining SAM Alerts.....	18
Defining MRM Alerts.....	19
Setting up Tivoli NetView for OS/390 and Source/390.....	26

Commands Support	26
Chapter 5. Setting Up the TBSM Server.....	27
Registry Modifications.	27
SQL Server Job Configuration	27
Defining the CICSplex	28
Aggregation	28
Aggregation Rules.....	29
Chapter 6. Event Processing Scenarios	31
CMAS Reports a Short on Storage State	31
CMAS Reports Short a Storage State has been Corrected.....	31
Chapter 7. Using CICSplex SM Instrumentation Tasks.....	33
Available Tasks	33
Chapter 8. Troubleshooting.....	35
Diagnostic Activities	35
Problem Determination	36
Appendix A. CICSplex Class Definitions.....	39
Appendix B. CICSplex Class Attributes.....	41
Appendix C. CICSplex Class States.....	43
Appendix D. IHSCCPPD Messages	45
Appendix E. Creating REXX Function Package Aliases.....	47
Appendix F. AOP Input Parameters.....	49
Continuing Parameters.	49
Parameters.....	50
Appendix G. TBSM Source/390 Object Pump Modify Commands.....	55
Appendix H. Using the NetView IHSCGLBL Command	59

1

Overview of CICSplex SM

In the past, most organizations could satisfy their data-processing needs with a single computer. A single CICS system could be used to handle their entire online transaction processing workload. Later, increased development in hardware and operating system technology, the need for more robust availability monitoring and high capacity transaction processing systems created the need for multiple CICS systems, which led to the concept of a CICSplex.

The term CICSplex (a CICS complex) is commonly used to describe two different concepts. Traditionally, the term CICSplex defines an environment in which two or more CICS regions are linked through CICS intercommunication facilities. Secondly, a CICSplex in conjunction with CICSplex System Manager, IBM's system management product for CICS networks, describes the largest set of CICS regions that can be manipulated as a single CICSplex entity.

The implementation of a CICSplex allows the CICS system to grow, but not without potential problems. Just as with a single CICS Region, the system programmer now has to consider the following issues:

- Effective use of resources
- Multiple operation points
- Increased difficulty in keeping track of what is happening

To bridge the gap between these issues, CICSplex SM, a subsystem monitor, was developed to automate the operations of CICS systems. CICSplex SM is the cornerstone for these automation processes because it reduces the complexity and in turn simplifies the management of complex collections of CICS systems by presenting them as an integrated whole. CICSplex SM manages CICS Systems under various Operating Systems (MVS/VSE/OS2). It provides services in the following areas:

- Resource management—an operator interface for complicated configurations that could include tens or hundreds of CICS regions.
- Workload management—provides intelligent routing mechanisms. It assigns incoming transactions according to the requirements and policies of the installation to the appropriate managed CICS regions.
- Resource monitoring—performs system availability and resource health checks automatically, generating alerts for further, possible automated, operator processing when it detects an abnormal situation.

CICSplex SM provides a real-time single system image of multiple CICS systems that make up transaction processing environments in many organizations. It also enables you to manage by exception processing by automatically warning you of deviations from intended performance and workload management.

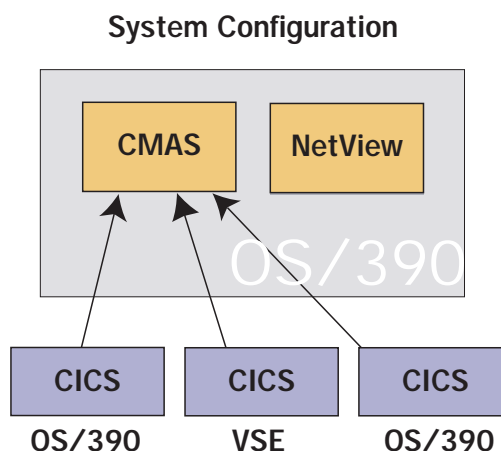
TBSM supports CICSplex SM by utilizing its exception source for CICS monitoring. When you define CICSplex SM resources within TBSM, you enlist its powerful monitoring and filtering capabilities to provide a business system-centric environment for control of these CICSplex SM objects. CICSplex SM passes events to TBSM via Tivoli NetView for OS/390. These exceptions are captured and posted to the TBSM workstation and appear as alerts related to specific objects. This interface enables TBSM to obtain event and real-time analysis information from CICSplex SM and all of its resources.

Once you have decided which CICS regions in your Enterprise you want to be managed by CICSplex SM, you must next ascertain how many CICSplexes to define to CICSplex SM and which CICS regions belong to each CICSplex. Your choice of single or multiple CICSplexes should mirror your current Enterprise organization. For example, a CICSplex managed by CICSplex SM could include every CICS region in your Enterprise or you could define multiple CICSplexes, each of which would consist of logical groupings of CICS systems. For further information on configuring CICSplex SM refer to the CICSplexSM Components and Planning Guide.

An important component of the CICSplex SM topology is the CICSplex SM Address Space (CMAS). It is responsible for the majority of work involving managing and reporting on CICS regions and their resources. A CMAS provides the single-system image for a CICSplex by serving as the interface to other CICSplexes and external programs

Each CICS system that is being managed by CICSplex SM is known as a Managed Application System (MAS). The MAS contain the agent code that runs as CICS Transactions and implements the various functions of CICSplex SM, such as data collection for the CMAS. A MAS can be local (LMAS) or remote (RMAS) depending on the communication link with its CMAS. A LMAS is on the same MVS image as the CMAS, while the RMAS resides on a different image than its CMAS. The following figure illustrates a typical CICSplex environment with Tivoli Netview for OS/390 (a prerequisite for integration) running on an OS/390 image.

Note: The level of detail reported on by CICSplex SM is far less for an RMAS than a LMAS.



2

Overview of TBSM Processing

Tivoli Business Systems Manager (TBSM) operates through two important processes. Objects to be managed need to be registered into the TBSM NT server database. Object discovery involves running batch jobs that detect the configuration and register these objects. Event processing involves capturing specific events and routing them into the TBSM Server that results in updates to the TBSM graphical user interface. These two important processes are described in the following sections, which allow you to understand the overall sequence as well as key facilities that are used.

Object Discovery

The object discovery and registration process insures that the TBSM model contains an accurate topology. The object discovery process involves the submission of a batch function. It uses the CICSplex SM Application Program Interface (API) to extract the information from the CICSplex SM Data Repository definition file in order to discover the CICSplex SM objects for TBSM. The batch module loops through the discovery process and creates a sequential file. This file is forwarded to the TBSM Server where it is processed into the SQL database.

Prediscovery jobs are run initially when TBSM is installed and customized. These jobs can be run on a scheduled basis or via operator intervention to capture updates to your configuration.

Object Rediscovery

Within a CICSplex, rediscovery occurs as a result of incoming event data that Tivoli Business Systems Manager is processing for availability. These events determine which CICSplex a particular resource may reside within as well as which Operating System the resource is running on and finally what the status is of the particular resource. If it is discovered that the resource has moved from one image to another, the database is modified to reflect the change. It appears as if the resource moved from one branch of the tree to another on the workstation. This is also true of other attributes that are maintained within the object model for that resource; for example, files and transactions.

Processing Events

TBSM provides event monitoring support for specific conditions as they occur on CICSplex SM managed resources, (CICSplex, CMAS, MAS, Transactions and Files). These CICSplex elements or resources are represented as objects on the Tivoli Business Systems Manager's graphical user interface.

The events collected by CICSplex SM are processed by TBSM in the same manner events from other sources such as the MVS console, and other data sources. Once passed from CICSplex SM, TBSM's Source /390 component forwards the events to a Windows NT based set of services, which creates event objects within the Tivoli Business System Manager data store. These newly created event objects are then linked to the objects that represent the resources where the condition occurred that ultimately generated the event.

The TBSM interface with CICSplex SM incorporates CICSplex SM's Real-Time Analysis component (RTA) to generate alerts based on a certain CICS resources moving to an unexpected state. These state changes are transmitted as generic SNA alerts to the Tivoli Netview for OS/390 Message Automation Table (MAT). To accomplish this a CMAS must be installed on the same MVS image as the Tivoli NetView for OS/390 instance. The events are passed through the program to program interface (PPI) and through the SNA session to TBSM for the instant visualization impact performance of the CICSplex object on your Enterprise. For example, CICSplex SM detects a Short on Storage condition in a CICS region. It issues an alert to Tivoli NetView for OS/390 (NetView) when it detects these events and NetView forwards it to Source/390, the OS/390 based TBSM component used for collecting OS/390 based events. The alert data is formatted and then forwarded for further processing by the Windows NT based Tivoli Business Systems Manager components. You view the event data on the Tivoli Business Systems Manager workstation.

Note: The CMAS must be installed on the same MVS image as NetView.

Event Flow

Event flow is as follows:

1. CICSplex SM employs the RTA function to indicate that one or more CICS components have changed state and have affected a CICS resource. CICSplex SM provides automated exception reporting in the form of a generic SNA alert to NetView.
2. Tivoli NetView for OS/390 processes these generic SNA alerts as Management Service Units (MSUs) in the Message Automation Table, which serves as a combination filter and dispatcher.
3. When the Message Automation Table detects a CICSplex SM event, it determines if it's of interest by examining its contents. If it is of interest, it invokes a module with either ALERT or RESOLVE as a parameter, depending on whether the event indicates a situation (for example, Short on Storage) or a resolution of an existing situation. The module formats the event and sends it to Source/390, the OS/390 based TBSM component through the Program to Program Interface (PPI).
4. The PPI Listener forwards the message to the TBSM Source/390 Dataspace and through to the TBSM Source/390 Object Server queue.
5. Source/390 then forwards the message to the Tivoli Business Systems Manager's NT Server environment.

The events that CICSplex SM generates are:

Exception	Description
SAMMXT	Maximum tasks
SAMSDM	SVC Dump
SAMSOS	Short on storage

Exception	Description
SAMTDM	Processing a transaction dump
SAMSTL	Stalled or suspended task
SAMOPS	CICS component availability
RTA	Real Time Analysis monitors for transactions and files

The events that are generated by CICSplex SM are categorized as Exceptions within Tivoli Business Systems Manager. Exceptions are events that are transient in nature and typically reflect a condition of duress. In contrast, Messages are events that reflect the Current State of a resource such as Starting, Active or Inactive. CICSplex SM generates these exception events when a condition occurs on a monitored CICSplex resource such as a Short on Storage, which is the SAMSOS exception in the preceding table. The exception is linked to the appropriate object representing the resource that generated it. When and if the condition is no longer valid a second exception is generated with a field in the exception set to the RESOLVE value. This indicates that the condition is no longer valid and negates or clears the previous condition.

Command Execution

TBSM provides query commands for a TBSM operator to determine information regarding CICS regions. The TBSM Server that contains the task server communicates with Tivoli NetView for OS/390 over a NETCONV session. This session can be a TCP/IP session or an LU6.2 session between TBSM and Tivoli NetView for OS/390. Upon selection of a CICS Region, the TBSM user interface displays a list of commands that are supported. The operator can select these commands, which run within the Tivoli NetView for OS/390 environment. Responses to the command are captured and displayed on the TBSM client.

3

Installation

The first step is to install the CICSplex SM CLISTS, samples, messages and JCL that enable discovery, event, and command processing for CICSplex SM. These data sets are installed through the System Modification Program Extended (SMP/E).

SMP/E Install

Install the CICSplex SM support using the directions in the Program Directory for Tivoli Business Systems Manager. Return to this book after installation for additional customization steps.

After SMP/E completes the TBSM installation, the target libraries contain three data sets that have the following low-level identifiers:

SGTMSAMP

contains SMP/E job control statements (JCL) samples, the Tivoli NetView for OS/390 message automation table and CICSplex SM batch definition files

SGTMEXEC

contains Tivoli NetView for OS/390 REXX CLISTS and message files

SGTMMODS

contains the discovery module IHSCPPD

Customizing Tivoli NetView for OS/390 Components

To customize Tivoli NetView for OS/390 Components:

1. Customize or add a Tivoli NetView for OS/390 message automation table for CICSplex SM.

If you have a message automation table currently in use in Tivoli NetView for OS/390, add the following statement to that message automation table:

```
%INCLUDE IHSCMAT1
```

The IHSCMAT1 member contains the CICSplex SM procedure statements needed for the message automation table

If you do not currently use a message automation table, refer to the Tivoli NetView for OS/390 Automation Guide for information about installing message automation tables

2. Include the GTMCMD member in DSICMD by adding the following statement to DSICMD in Tivoli NetView for OS/390:

```
%INCLUDE GTMCMD
```

The GTMCMD member defines the necessary parameters for the CICSplex SM instrumentation.

-
3. Concatenate the SGTMSAMP data set to the data definition (DD) statement for DSIPARM in the Tivoli NetView for OS/390 start procedure.

The SGTMEEXEC data set contains message files and Tivoli NetView for OS/390 CLISTs, all of which are new members for CICSplex SM instrumentation. Perform the following:

1. Copy the CICSplex SM Instrumentation message files, DSIIHS00 and DSIIHS01, from the SGTMMMSGs data set to the data set specified in the DD statement for DSIMSG in the Tivoli NetView for OS/390 start procedure.
2. Concatenate the SGTMEEXEC data set to the DD statement for DSICLD in your Tivoli NetView for OS/390 start procedure statement to add Tivoli NetView for OS/390 CLISTs for CICSplex SM instrumentation.

Installing the REXX Function Package

The CICSplex SM application program interface (API) REXX function package is used with the Tivoli NetView for OS/390 address space. The REXX runtime interface to the API is shipped as a function package and host command. This interface consists of a single load module that contains the following entry points:

EYU9AR00

The function package, which is shipped in the SYS1.CPSMnnn.SEYUAUTH library with an alias of IRXFLOC, where nnn is your current release.

EYU9AR01

The host command

For a REXX program to access the function package, the module EYU9AR00, along with its alternate entry point, EYU9AR01, and its alias, IRXFLOC, must reside in an authorized library in one of these places:

- The OS/390 linklist
- The STEPLIB concatenation of the application that calls the API

For a REXX program to access the function package from Tivoli NetView for OS/390, the EYU9AR00 module must also be aliased to DSIRXFLP and placed in an authorized library in either the OS/390 linklist or the STEPLIB concatenation for the Tivoli NetView for OS/390 system.

Note: The REXX runtime interface is a CICSplex SM API application, and must have appropriate access to the EYU9AB00 module.

SMP/E Control Statements

The following members contain SMP/E user modification control statements that you can use to move the necessary API load modules to the SYS1.CPSMnnn.SEYULINK library when nnn is your CPSMnnn.SEYUINST.

Member	Load Module
EYU\$UM11	EYU9AR00
EYU\$UM12	EYU9AB00
EYU\$UM13	EYU9XESV

Alias Considerations

When you use the IRXFLOC or DSIRXLFP aliases to provide access to the REXX function package to avoid conflicts with other Tivoli NetView for OS/390 function packages installed for your enterprise, you must place them ahead of any other IRXFLOC or DSIRXLFP modules in the STEPLIB (or OS/390 linklist) concatenation. See “Creating REXX Function Package Aliases” in Appendix G for details.

When you do not use the aliases for the REXX function package, you must modify your REXX parameter modules (IRXPARMS, IRXTSPRM, and IRXISPRM).

Note: We recommend that you add the function package shipped with CICSplex SM as systems function package rather than a local or user function package.

Host Command Entry

Add a host command entry for the REXX function package in the following format:

- An 8-byte command environment name of CICSplex
- An 8-byte command routine name of EYU9AR01
- A 16-byte command token of blanks

The final step to complete installation of the REXX function package is to increase the number of entries in the appropriate function package table and add an entry to that table for EYU9AR00.

4

OS/390 Customization

Information in this chapter describes the configuration of CICSplex SM that is required for integration with TBSM.

Setting up the Discovery Job

TBSM discovery for CICSplex SM is accomplished by running IHSCCRUN– a two step job. The first step is to execute IHSCCPPD. This step extracts information about CICSplex objects for use in TBSM. It builds a sequential file with variable length records where each record represents one discovered object.

The second step is to execute GTMAOPE0, which transmits the sequential file to the NT Server that contains the TBSM database. Executing GTMAOPE0 is discussed later in this chapter.

IHSCCPPD uses the CICSplex SM Application Programming Interface (API) to extract information about CICS objects. The objects are from the following categories:

CMAS

CICSplex Management Address Space – These are CICS systems dedicated to the CICSplex SM function. Normally they do not run business transactions. They are not associated with any particular CICSplex and may supply information to multiple CICSplex SM environments.

MAS Managed Address Space – These are CICS systems performing user function, i.e. normal business transactions. Their status is reported via a CMAS, either local or remote.

LOCTRAN

Local Transaction – These are transactions that reside on a MAS.

REMTRAN

Remote Transaction – These are transactions that may be invoked on one MAS but actually reside on another MAS.

LOCFILE

Local File – These are files that reside on a MAS.

REMFILE

Remote File - Files that may be accessed on one MAS but actually reside on another MAS.

The CMAS and MAS objects may have several states. ACTIVE indicates the MAS/CMAS has completed initialization and is in a normal running state. INACTIVE indicates the MAS/CMAS is not present in the system. The other states represent transient conditions.

Active objects provide much more information than those objects in other states. For example, transactions and files information is only available from active objects.

Note: If a CICS Region is active, it is known to be active by the CMAS that manages it plus any CMASs that have direct or indirect connections to that CMAS. If a CMAS is active, it is known to be active by itself plus any CMASs that have direct or indirect connections to it. Therefore, when you run the discovery job, you want to connect it to a CMAS that has connectivity to all other CMASs in the CICSplex, if possible. Normally, this would be the Maintenance Point CMAS. If you connect to a CMAS that does not have this connectivity, the discovery will miss objects or incorrectly identify active objects as inactive.

Using the SubName Capability

In some TBSM environments, it is possible to have multiplex CICSplexs with the same name. Although this is not allowed in a CICSplex domain, it is possible to have multiple CICSplex domains within a single TBSM environment. The SubName capability allows you to add a unique SubName to each of the duplicate CICSplex names for identification purposes. This SubName appears as an append to the CICSplex name, separated by a dot. The SubName capability function is transparent to CICSplex SM, which continues to use the original, duplicated CICSplex names.

If you use the SubName capability the following restrictions apply:

- All ACTNDEFs for the CICSplex that has a SubName must be defined on a single CMAS. This is referred to as the reporting CMAS and is usually the Maintenance Point CMAS.
- You must run the Discovery Job for a CICSplex on the MVS image that contains the reporting CMAS for that CICSplex. This MVS image must support a NetView environment and a Source/390 (Object Pump, DataSpace, and Object Server) environment.
- Installation functions pertaining to NetView (such as the Command List entry, the Automation Table entry, and the NetView modules) must be installed before running the Discovery Job.
- The NetView system should be running when you run the Discovery job. If the NetView system is not running, you must enter the SubName information manually using the IHSCGLBL NetView command. Using this command is described in Appendix J of this book.

Defining the SubName

Define the CICSplex to TBSM with a name of the form ccccccc.sssssss where ccccccc is the CICSplex Name as defined to CICSplex SM and sssssss is the SubName you're adding.

To define the SubName, perform one of the two following steps:

- In the IHSCPPD step of the Discovery Job, you may define the SubName as a part of the CICSplex Name in the form ccccccc.sssssss where ccccccc is the CICSplex Name as defined to CICSplex SMCICSplex SM and sssssss is the SubName being added to it.

- Enter the NetView command `IHSCGLBL CICSplexName=ccccccc SubName=sssssss` where ccccccc is the CICSplex Name as defined to CICSplex SM and sssssss is the SubName being added to it. (See the explanation of the IHSCGLBL command in Appendix J of this book.)

If a SubName is defined for a CICSplex, it appears for all representations of that CICSplex object as well as all representations of any CICS objects that are associated with that CICSplex.

Creating the Discovery Job

Job Statement

The job statement should meet installation standards.

Extract Step

EXEC Statement

```
//EXTRACT EXEC PGM=IHSCPPD,
PARM='CICSplex Name,CMAS Name,Security Value,USERID'
```

CICSplex Name

The 1-8 character name of the CICSplex for which the discovery is being done. If the installation supports multiple CICSplexes, it will be necessary to run multiple extractions. If SubName support is desired for this CICSplex, use the 3-17 character name of the form CICSplexName.SubName.

CMAS Name

The 1-8 character name of the CMAS on the OS/390 image to which IHSCPPD is to connect. This should be the Maintenance Point CMAS or a CMAS that has connectivity to the other CMASs involved in the management of the CICSplex.

Security Value

If your CICSplex installation has the API security exit EYU9XESV, this is the 1-8 character SIGNONPARM (Signon parameter) which will be passed to the CICSplex and to that security exit.

USERID

If your CICSplex installation has the API security exit EYU9XESV, this is the user 1-8 character User ID which will be passed to the CICSplex and to that security exit.

An example follows.

```
//RUN EXEC PGM=IHSCPPD,
// PARM='CICSPLX1,CMAS01,SECVL1,0130,USER1'
```

STEPLIB DD Statement

At run time, CICSplex SM must find the module EYU9AB00 in the STEPLIB, MVS linklist, or LPA library concatenation. This module is distributed in the CPSPMnnn.SEYAUTH library where nnn is the version of CICSplex SM. In the example that follows, IHSCPPD is in the library hlq.SGTMMODS, and EYU9AB00 is in the library hlq.CPSPMnnn.SEYUAUTH:

```
//STEPLIB DD DISP=SHR,DSN=CPSPMnnn.SEYUAUTH
// DD DISP=SHR,DSN=hlq.SGTMMODS
```

SYSPRINT DD Statement

IHSCCPPD produces messages on the SYSPRINT DD data set which is normally allocated to JES.

```
//SYSPRINT DD SYSOUT=*
```

AOPDFILE DD Statement

IHSCCPPD writes to a file with DDNAME AOPDFILE. It can create a new file or add to an existing file. IHSCCPPD defines the following parameters for the file:

- RECFM - VB
- BLKSIZE – 4096
- LRECL – 4092

The following example creates a new file.

```
//AOPDFILE DD DISP=(NEW,CATLG),DSN=AOP.DISCOVERY,  
//          VOL=SER=USRVOL,SPACE=(TRK,(10,1)),UNIT=3390
```

The following example adds to an existing file:

```
//AOPDFILE DD DISP=(MOD,KEEP),DSN=AOP.DISCOVERY
```

The majority of records written to the file are approximately 110 bytes in length. The maximum record length is approximately 130 bytes. The total number of records is the sum of the following:

- The number of CMAS regions in the CICSplex
- The number of MAS regions in the CICSplex
- For each active MAS region:
 - The number of local transactions defined to the MAS.
 - The number of remote transactions defined to the MAS.
 - The number of local files defined to the MAS.
 - The number of remote files defined to the MAS.

Transmit Step

The GTMAOPE0 utility is used to read any non-VSAM file and seamlessly transmit the data to TBSM. GTMAOPE0 is a standalone MVS TCP/IP Application that can send data to the TBSM Server running the ASIMVSIPLListenerSvc.

GTMAOPE0 CONTROL CARDS

TCPIP_ADDRESS=nnn.nnn.nnn.nnn
TCPIP_NAME=hostname

Specifies the IP address number or hostname for the connection that is used to communicate with TBSM server running ASIMVSIPLListenerSvc. These are required operands. However, only one of these parameters is required to be specified. See the Tivoli Business Systems Manager Installation and Configuration Guide for details on how to install the ASIMVSIPLListenerSvc.

If the TCPIP_NAME parameter is used, GTMAOPE0 attempts Host Name Resolution to locate the IP address. The hostname is alphanumeric and the length can range from 1 to 69 characters long.

If the TCPIP_ADDRESS parameter is used, the address is the 4-octet internet address of the NT Server upon which the ASIMVSIPLListenerSvc resides. This can be obtained by issuing the IPCONFIG command from a command prompt on the NT Server.

If traceresolver is set on in the TCP/IP parameters, TCP/IP debug messages are issued to the job log when host name resolution is attempted. These messages are issued regardless of whether the IP Address is found.

The resolvertimeout and resolverudpretries parameters of TCP/IP control the time specifications for Host Name Resolution processing within the TCP/IP network.

The TCP/IP address or hostname is defined in the Windows/NT registry of the machine running the ASIMVSIPLListenerSvc under the ValidClients key. See “Setting Up the TBSM Server” section for more information.

Refer to the TCP/IP manuals for more information regarding the TCP/IP parameters described here.

TCPIP_PORT=nnnnn

Specifies the port number that the ASIMVSIPLListenerSvc is using. See the Tivoli Business Systems Manager Installation and Configuration Guide for details on how to install this service. The required operand nnnnn can be in the range from 1 to 32767.

The TCP/IP port address is defined in the Windows/NT registry on the machine running the ASIMVSIPLListenerSvc under the Settings key. See “Setting Up the TBSM Server” section for more information.

TCPIP_JOBNAME=tcip_jobname

Specifies the TCP/IP address space that is to be used to provide TCP/IP services. The TCP/IP Jobname cannot exceed 8 characters in length. When not specified, TCP/IP dynamically locates the TCP/IP address space to use. This is an optional operand.

TIMEOUT=30 | seconds

Specifies the timeout value in seconds in which the TCP/IP services terminates if there is no response. The default is 30 seconds. The maximum value is 86400 seconds. This is an optional operand.

This parameter works in conjunction with the TCP/IP resolvertimeout parameter. For example, when the value of this parameter is less than the resolvertimeout parameter, the IP service requests time out prior to TCP/IP terminating the outstanding service request.

Refer to the TCP/IP manuals for more information regarding the resolvertimeout TCP/IP parameter.

CODEPAGE=codepage_number | 037

The codepage number corresponding to the OS/390 you are transmitting data from. For U.S. EBCDIC the codepage is X'37'. This codepage number to be used by the ASIMVSIPLListenerSvc to convert the data to the appropriate ASCII symbols. This is an optional control card.

FORMAT=format_code | x'00'

Specifies a format data type used to identify the type of data being Transmitted to TBSM. This is an optional operand. When not specified the format value received by TBSM is x'00'. A decimal value may be specified to identify the format code. The range is from 1 to 255. For CICSplex SM, the format code is X'13' or decimal 19.

COMMAND=command_name

Specifies a command name sent to the ASIMVSIPLListenerSvc for triggering the command execution on NT to process the data after all the data has been received. The Command_name is an alphanumeric string that can range from 1 to 72 characters long. This is a required operand.

This command is defined in the Windows/NT Registry on the machine running the ASIMVSIPLListenerSvc, under the CommandAliases key. See "Setting Up the TBSM Server" section for more information. For CICSplex SM this value is CICSDiscovery

USERDATA=userdata_string

Specifies a user string that is sent to the ASIMVSIPLListenerSvc. This string is used for additional validation of the transmission process. Userdata_string is an alphanumeric string that can range from 1 to 71 characters long. This is an optional operand.

Note: '*' in col. 1 and blank lines are treated as comments.

Discovery Job Example

Following is an example of the execution JCL for a CICSplex SM discovery run.

```
//IHSCCRUN JOB ,CLASS=A,MSGLEVEL=1
//      SET CPSMLIB=dfh.cpsm      HIGH LEVEL QUALIFIER FOR CPSM LIBRARY
//      SET TBSMLIB=tme.tbsm      HIGH LEVEL QUALIFIER FOR TBSM LIBRARY
//*****
//*
//*  Licensed Materials - Property of IBM
//*  5698-BSM (C) COPYRIGHT IBM Corp. 2001
//*
//*  5698-BSM (C) COPYRIGHT Accessible Software 1999
//*  ALL RIGHTS RESERVED
//*
//*  US GOVERNMENT USERS RESTRICTED RIGHTS
//*  - USE, DUPLICATION OR DISCLOSURE RESTRICTED BY
//*  GSA ADP SCHEDULE CONTRACT WITH IBM CORPORATION
//*
//*
//*****
//* Change Activity:
//* CFD List:
//* $A0=PTM00207      V1R1      002808 ROS      : Formed
//* $P1=PTM00297,TIVMG12,03/22/01, DK: Copyright changes
//* $P2=PTM00323,TIVMG12,04/03/01, DE: Module rename
//* $P3=PTM00348,TIVMG12,04/26/01, ROS: Simplify and correct
//* End CFD List:
```

```

/* Additional Notes about the Change Activity                                *
/* $xx Optional Additional Text: See Automatic Change Flagging              *
/* End Change Activity:                                                    *
/******                                                                    *
/*
/*
/* THIS STEP EXTRACTS THE INFORMATION FROM CPSM AND CREATES THE
/* SEQUENTIAL FILE
/*
/*EXTRACT EXEC PGM=IHSCCPD,PARM='TIVOLI,NTA7C01,BOBS,0130,USER4'
/*STEPLIB DD DISP=SHR,DSN=&CPSMLIB..SEYUAUTH
// DD DISP=SHR,DSN=&TBSMLIB..SGTMMODS
//SYSPRINT DD SYSOUT=*
//AOPDFILE DD DISP=(NEW,PASS,DELETE),DSN=&&TEMP,
//          SPACE=(TRK,(10,1)),UNIT=SYSDA
/*
/*
/* THIS STEP TRANSMITS THE FILE TO THE NT SERVER
/*
//TRANSMIT EXEC PGM=GTMAOPE0 @P2C
//STEPLIB DD DSN=&TBSMLIB..SGTMMODS,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=*.EXTRACT.AOPDFILE,DISP=(OLD,DELETE,DELETE)
//SYSIN DD *
TCPIP_ADDRESS=address_of_SQL_server
*TCPIP_NAME=name_of_SQL_server
TCPIP_PORT=1021
CODEPAGE=037
*BUFFERSIZE=3
*TIMEOUT=90
COMMAND=CICSDISCOVERY
*USERDATA=user_string
CONVERT=NO
*TEXT=YES
DATA_SEPARATOR=x'25'
FORMAT=X'13'
/*
//

```

Discovery Job Sample Output

Following is a sample of a successful execution of the IHSCCPD job. There were no remote transactions or files for the 1 active MAS object.

```

IHSCCPD CICSPlex Name for this run is TIVOLI
IHSCCPD CMAS Name for this run is NTA7C01
IHSCCPD Security value for this run is BOBS
IHSCCPD CPSM Release for this run is 0130
IHSCCPD UserID for this run is USER4
IHSCCPD Total number of CMAS objects is      1
IHSCCPD Total number of active CMAS objects is      1
IHSCCPD Total number of MAS objects is      4
IHSCCPD Total number of active MAS objects is      1
IHSCCPD Processing records for MAS NTA7A01
IHSCCPD Total number of records is      143

```

GTMAOPE0 Job Sample Output

```

GTM8040I  DATA TRANSMISSION STARTED
GTM8041I  SYSUT1 RECORDS SENT -          143
GTM8030I  PROCESSING COMPLETED SUCCESSFULLY

```

CICSPlex SM Definition for TBSM

The following section describes the configuration of CPICSPlex SM required for integration with TBSM.

TBSM uses the Real Time Analysis (RTA) feature of CICSplex Systems Management. This feature is described in these CICSplex SM publications:

- GC33-0786 CICSplex SM Concepts and Planning – Overview of the function
- SC34-5401 CICSplex SM Administration – Use of the CICSplex SM panels
- SC33-1808 CICSplex SM Managing Resource Usage - Configuration

Activities that are necessary to use CICSplex SM in the TBSM environment are discussed in the following sections. For detailed descriptions of CICSplex SM, refer to the publications listed above.

CICSplex SM Objects Required for TBSM

TBSM requires the following definitions of CICSplex SM objects:

CICSSYS

Defines the CICS system and associates it with a CICSplex.

RTASPEC

Analysis specification that identifies the default control attributes that are used for system availability monitoring (SAM) and provides an anchor for all analysis definitions and status definitions associated with a CICS system. For TBSM it is the anchor for the RTAGROUP specification that points to the RTADEF specifications that are used for MRM (MAS Resource Monitoring).

RTADEF

Analysis definition that defines the evaluations to be performed on a periodic basis and the actions to be taken should a notifiable condition occur. For TBSM, this is used to define transaction and file status reporting. An RTADEF will have a related EVALDEF to define the evaluation and a related ACTNDEF to define the action.

EVALDEF

Evaluation definition that identifies the resources that are to be sampled and evaluated. When the result of the evaluation is true, the associated analysis definition (RTADEF) is used to determine if a notifiable condition has occurred.

ACTNDEF

Action definition that designates the type of external notification that is to occur when the condition or conditions identified in the analysis definition are true.

Defining SAM Alerts

For SAM alerts, the analysis specification (RTASPEC) definition specifies action definitions (ACTNDEFs) to be used for the SAM alerts. Thus, for SAM alerts, there are no analysis definitions (RTADEFs) or evaluation definitions (EVALDEFs). There are default action definitions (ACTNDEFs) provided for the SAM alerts but they are not correct for TBSM use.

SAM defines six conditions and TBSM will handle all of them. However, the user may not want to use all of them depending on the operating environment. The six conditions are:

SAM System Unavailable

SOS Short on Storage

SYSDUMP

System dump in progress

TRANDUMP

Transaction dump in progress

MAXTASK

System at maximum number of tasks

STALL

System in stall condition (deadlock)

To set up for SAM conditions:

1. Use the CICSplex SM interactive panels and enter RTASPEC. This will provide a list of RTASPEC objects defined for your CICSplex.
2. Find the RTASPEC object corresponding to the MAS (CICS system) for which you are doing the setup. Enter UPD as the action for this RTASPEC.
3. Determine which of the six conditions you wish to generate alerts. For those conditions, specify an action definition (ACTNDEF) in the ACTION field. For CICSplex SM purposes, the severity is not used. The ACTNDEF may be an existing definition; however, if it is used for other purposes, make sure that usage is compatible with usage for TBSM.
4. For the ACTNDEFs specified above, ensure the following items are specified:
 - ALERT is set to YES
 - CMAS NAME is specified as the local CMAS; preferably, the CMAS that is sending alerts directly to NetView.

This may be done by specifying ACTNDEF on the CICSplex SM interactive panels and then either DEF or UPD for each ACTNDEF; depending on whether you are defining new ones or modifying old ones.

Defining MRM Alerts

The MAS Resource Monitor (MRM) facility is used to obtain information about transactions and files. The objects of interest are the analysis definition (RTADEF), the evaluation definition (EVALDEF) and the action definition (ACTNDEF). These definitions are related to the MAS by an analysis group (RTAGROUP) definition and by analysis/status definitions in analysis group (RTAINGRP) definitions.

MRM operates by periodic sampling of the state of a resource (file or transaction). To understand this and the relationship of the three object definitions, consider the following scenario:

1. A resource (file or transaction) is to be monitored. The analysis definition (RTADEF) defines the frequency of the monitoring activity in seconds using the Sample Interval specification.
2. When the sample interval has expired, CICSplex SM uses the evaluation expression in the analysis definition to determine what to check. This expression may consist of one or more evaluation definitions (EVALDEFs) connected by a set of logical operators.
3. The EVALDEF specifies the type of resource in the Tablename specification. For TBSM purposes, this should be LOCTRAN, REMTRAN, LOCFILE or REMFILE.
4. The EVALDEF specifies the specific resource or family of resources to be evaluated using the Instance Pattern specification. This is compared to resource names as defined

in CICS. For example, a Tablename value of LOCTRAN and an Instance Pattern of “IHS*” would produce all transactions whose names begin with “IHS”.

5. The EVALDEF specifies the evaluation criteria via the Evaluation Column, Value and Value specifications. For TBSM, these should have one of the two following formats:
 - To test files for open/close status – OPENSTATUS NE OPEN
 - To test files or transactions for enable/disable status – ENABLESTATUS NE ENABLE
6. There are multiple reasons for a file not to be enabled. TBSM will handle all of these conditions if the CICSplex SM is set up correctly. There must be a separate EVALDEF, RTADEF and ACTNDEF for each condition. See the table below for correct setup.
7. The EVALDEF specifies the Severity of the event. This is used in the next step.
8. For each severity, the Entry Intervals field in the analysis definition (RTADEF) object specified the number of consecutive evaluation time periods the designated condition or conditions must be true before any action is taken. Thus a user could specify that with an evaluation time period of 1 minute, a low-severity condition must be true for 5 minutes before an alert is generated.
9. For each severity, the Exit Intervals field in the analysis definition (RTADEF) object specified the number of consecutive evaluation time periods the designated condition or conditions must be false before any action is taken. Thus a user could specify that with an evaluation time period of 1 minute, a high-severity condition must be false (after being true) for 5 minutes before an alert is removed.
10. Each analysis definition must have a related action definition (ACTNDEF). It must contain the following entries:

Name Action name specified in analysis definition

Alert Yes

CMAS Name

Name of CMAS which is forwarding alerts to NetView

Enter Text

See tables. that follow.

Exit Text

See tables that follow.

Tables of Values for MRM Alert Generation

The following tables describe values for MRM alert generation.

Transaction Enable/Disable Alerts

RTADEF

Name	User-specified name
Perform Ops	No
Sample Interval	User-specified interval. A small interval will improve responsiveness of TBSM while consuming more CPU time.
Action Name	Name of the related ACTNDEF object

Entry Intervals	The number of consecutive evaluation time periods, for each severity level, during which the designated condition or conditions must be true before any action is taken.
Exit Intervals	The number of consecutive evaluation time periods, for each severity level, during which the designated condition or conditions must be false before any action is taken.
Evaluation Expression	One or more evaluation definitions (EVALDEF'S) that are to be analyzed.

EVALDEF

Name	One of the names specified in the EVALUATION EXPRESSION in the analysis definition
Sample Interval	User-specified interval. A small interval will improve responsiveness of TBSM while consuming more CPU time.
Table Name	LOCTRAN/REMTRAN
Instance Pattern	A pattern used to recognize the name of resources to be evaluated. It may include a wildcard (*) as a part of the name.
Result Set Action	Any
Separate Task	NO (Unless sample interval is small enough to impact CICS performance)
Evaluation Column	ENABLESTATUS
Operation	NE
Value	ENABLED
Severity	User-specified
View Invoked	LOCTRAN/REMTRAN
Filter String Expression	None
Modification String	None

ACTNDEF

Action Name	Action Name specified in the RTADEF
Event	Yes
Event View	LOCTRAN/REMTRAN
Priority	User Choice
Event Description	Transaction Enable/Disable
Alert	Yes
CMAS Name	Name of CMAS receiving alert. It should be the CMAS which is communicating with NetView.
Enter Text	Disabled
Exit Text	Enabled
Restart	No

File Enable/Disable Alerts

RTADEF

Name	User-specified name
Perform OPS	No
Sample Interval	User-specified interval. A small interval will improve responsiveness of TBSM while consuming more CPU time.
Action Name	Name of the related ACTNDEF object
Entry Intervals	The number of consecutive evaluation time periods, for each severity level, during which the designated condition or conditions must be true before any action is taken.
Exit Intervals	The number of consecutive evaluation time periods, for each severity level, during which the designated condition or conditions must be false before any action is taken.
Evaluation Expression	One or more evaluation definitions (EVALDEF'S) that are to be analyzed.

EVALDEF

Name	One of the names specified in the EVALUATION EXPRESSION in the analysis definition
Sample Interval	User-specified interval. A small interval will improve responsiveness of TBSM while consuming more CPU time.
Table Name	LOCFILE/REMFIL
Instance Pattern	A pattern used to recognize the name of resources to be evaluated. It may include a wildcard (*) as a part of the name.
Result Set Action	Any
Separate Task	NO (Unless sample interval is small enough to impact CICS performance.)
Evaluation Column	ENABLESTATUS
Operation	NE
Value	ENABLED
Severity	User-specified.
View Invoked	LOCFILE/REMFIL
Filter String Expression	ENABLESTATUS EQ DISABLED
Modification String Expression	None

ACTNDEF

Action Name	Action Name specified in the RTADEF
Event	Yes
Event View	LOCFILE/REMFIL
Priority	User Choice
Event Description	File Enable/Disable
Alert	Yes

CMAS Name	Name of CMAS receiving alert. It should be the CMAS which is communicating with NetView.
Enter Text	DISABLED
Exit Text	ENABLED
Restart	No

File Enable/Unenable Alerts

RTADEF

Name	User-specified name
Perform Ops	No
Sample Interval	User-specified interval. A small interval will improve responsiveness of TBSM while consuming more CPU time.
Action Name	Name of the related ACTNDEF object
Entry Intervals	The number of consecutive evaluation time periods, for each severity level, during which the designated condition or conditions must be true before any action is taken.
Exit Intervals	The number of consecutive evaluation time periods, for each severity level, during which the designated condition or conditions must be false before any action is taken.
Evaluation Expression	One or more evaluation definitions (EVALDEFs) that are to be analyzed.

EVALDEF

Name	One of the names specified in the EVALUATION EXPRESSION in the analysis definition
Sample Interval	User-specified interval. A small interval will improve responsiveness of TBSM while consuming more CPU time.
Table Name	LOCFILE
Instance Pattern	A pattern used to recognize the name of resources to be evaluated. It may include a wildcard (*) as a part of the name.
Result Set Action	Any
Separate Task	NO (Unless sample interval is small enough to impact CICS performance.)
Evaluation Column	ENABLESTATUS
Operation	NE
Value	ENABLED
Severity	User-specified
View Invoked	LOCFILE/REMFIL
Filter String Expression	ENABLESTATUS EQ UNENABLED
Modification String Expression	None

ACTNDEF

Action Name	Action Name specified in the RTADEF
Event	Yes
Event View	LOCFILE/REMFIL
Priority	User choice
Event Description	File Enable/Disable
Alert	Yes
CMAS Name	Name of CMAS receiving alert. It should be the CMAS which is communicating with NetView
Enter Text	DISABLED
Exit Text	ENABLED
Restart	No

File Enable/Disabling Alerts

RTADEF

Name	User-specified name
Perform OPs	No
Sample Interval	User-specified interval. A small interval will improve responsiveness of TBSM while consuming more CPU time.
Action Name	Name of the related ACTNDEF object
Entry Intervals	The number of consecutive evaluation time periods, for each severity level, during which the designated condition or conditions must be true before any action is taken.
Exit Intervals	The number of consecutive evaluation time periods, for each severity level, during which the designated condition or conditions must be false before any action is taken.
Evaluation Expression	One or more evaluation definitions (EVALDEF'S) that are to be analyzed.

EVALDEF

Name	One of the names specified in the EVALUATION EXPRESSION in the analysis definition
Sample Interval	User-specified interval. A small interval will improve responsiveness of TBSM while consuming more CPU time
Table Name	LOCFILE
Instance Pattern	A pattern used to recognize the name of resources to be evaluated. It may include a wildcard (*) as a part of the name.
Result Set Action	Any
Separate Task	NO (Unless sample interval is small enough to impact CICS performance.)
Evaluation Column	ENABLESTATUS
Operation	NE
Value	ENABLED

Severity	User-specified
View Invoked	LOCFILE/REMFIL
Filter String Expression	ENABLESTATUS EQ DISABLING
Modification String Expression	None

ACTNDEF

Action Name	Action Name specified in the RTADEF
Event	Yes
Event View	LOCFILE
Priority	User Choice
Event Description	File Enable/Disabling
Alert	Yes
CMAS Name	Name of CMAS receiving alert. It should be the CMAS which is communicating with NetView
Enter Text	DISABLING
Exit Text	ENABLED
Restart	No

File Open/Close Alerts

RTADEF

Name	User-specified name
Perform OPs	No
Sample Interval	User-specified interval. A small interval will improve responsiveness of TBSM while consuming more CPU time.
Action Name	Name of the related ACTNDEF object
Entry Intervals	The number of consecutive evaluation time periods, for each severity level, during which the designated condition or conditions must be true before any action is taken.
Exit Intervals	The number of consecutive evaluation time periods, for each severity level, during which the designated condition or conditions must be false before any action is taken.
Evaluation Expression	One or more evaluation definitions (EVALDEF'S) that are to be analyzed.

EVALDEF

Name	One of the names specified in the EVALUATION EXPRESSION in the analysis definition
Sample Interval	User-specified interval. A small interval will improve responsiveness of TBSM while consuming more CPU time.
Table Name	LOCFILE
Instance Pattern	A pattern used to recognize the name of resources to be evaluated. It may include a wildcard (*) as a part of the name.

Result Set Action	Any
Separate Task	NO (Unless sample interval is small enough to impact CICS performance.)
Evaluation Column	OPENSTATUS
Operation	NE
Value	OPEN
Severity	User-specified.
View Invoked	LOCFILE
Filter String Expression	None
Modification String Expression	None

ACTNDEF

Action Name	Action Name specified in the RTADEF
Event	Yes
Event View	LOCFILE
Priority	User choice
Event Description	File Open/Close
Alert	Yes
CMAS Name	Name of CMAS receiving alert. It should be the CMAS which is communicating with NetView
Enter Text	CLOSED
Exit Text	OPEN
Restart	No

Setting up Tivoli NetView for OS/390 and Source/390

The Program to Program Interface (PPI) between Tivoli NetView for OS/390 and Source/390 is used to transport the events to the TBSM Server. Ensure that Tivoli NetView for OS/390 has the PPI option enabled. Upon starting the TBSM Source/390 Object Pump, the PPI receiver is running.

Commands Support

To execute commands between TBSM and Tivoli NetView for OS/390, you must enable either an LU 6.2 conversation or a TCP/IP connection using NETCONV. For more information on enabling command communication, refer to the Tivoli Business Systems Manager Installation and Configuration Guide.

5

Setting Up the TBSM Server

The following section describes the installation of the Windows NT-based components necessary for processing CICSplex discovery.

Registry Modifications

On hosts running the ASIMVSIPLListenerSvc, configure the appropriate entry in the Registry for the processing to be done. The following registry keys define the processing for the CICS discovery feed. This is configured by default.

The following setting defines the port address to communicate with GTMAOPE0. This is specified on the TCPIP_PORT control card of GTMAOPE0.

HKEY_LOCAL_MACHINE \ SOFTWARE \ Accessible Software,Inc. \ Access1 \ 1.0 \ Components \ ASIMVSIPLListenerSvc \ Settings \ Port 1021

The following setting defines the command to be triggered on the Windows/NT server by GTMAOPE0. This is specified on the COMMAND control card of GTMAOPE0.

**HKEY_LOCAL_MACHINE \ SOFTWARE \ Accessible Software,Inc. \ Access1 \ 1.0 \ Components \ ASIMVSIPLListenerSvc \ Settings \ CommandAliases
CICSDISCOVERY sh CreateDiscoveryBatch.ksh -F19 -A1 -C37 %s**

The following setting defines the IP clients that are authorized to run GTMAOPE0. This is specified on the TCPIP_ADDRESS or TCPIP_NAME control card of GTMAOPE0.

HKEY_LOCAL_MACHINE \ SOFTWARE \ Accessible Software,Inc. \ Access1 \ 1.0 \ Components \ ASIMVSIPLListenerSvc \ Settings \ ValidClients

SQL Server Job Configuration

Two SQL Server jobs are installed for CICSplex support and are used in the discovery of CICSplex resources. The CICS Discovery Load job takes any Discovery Batch that is in the ENQUEUED State and loads the associated file into the SQL Server database. The CreateDiscoveryBatch.ksh script defined in the registry (as shown above) creates the Discovery Batch, and assigns it a state of ENQUEUED. The TBSM administrator should use the SQL Enterprise Manager and define a schedule for this job to run. This is a polling based job and should run on intervals of several minutes during the course of a window of time that is designated. If there is no CICS Discovery Batch that is in the ENQUEUED state, the job simply logs a message and exits. If it does find a Discovery Batch in the

ENQUEUED state, and the job is successful the Discovery Batch will be marked as LOADED. A failure causes the Discovery Batch to be marked as being in the LOAD_ERROR state.

The CICS Discovery Process job performs the processing required on any CICS Discovery Batch that is in the LOADED state. Like the CICS Discovery Load job, the TBSM administrator has to define a schedule for this job. When a Discovery Batch is processed it can place significant load on the database, therefore we recommend you schedule the job for low-activity periods.

Defining the CICSplex

For CICSplex managed objects to be discovered, a CICSplex must be defined to TBSM. This is achieved by using the TBSM workstation and inserting a CICSplex object from any one of the following objects:

- Enterprise
- Complex
- Machine
- LPAR
- Operating System

Once the CICSplex is defined into TBSM, the discovery data generated in the OS/390 environment is processed and used to populate the TBSM repository. Objects such as CICS regions, CMAS, Files, and Transactions are discovered and inserted on the Enterprise Outliner or Business Object Container.

Aggregation

The Business Object Container folder (BUSC) commonly referred to as the Enterprise Outliner represents resources as objects within a hierarchical view that reflects where that resource belongs in the physical world. For example, Operating Systems run on machines which are located within a Complex, which are within an Enterprise. TBSM requires that each resource being monitored must have an object representation within the Enterprise Outliner. Either this can be accomplished through various auto discovery methods where systems are interrogated to find out what is running on them, or specialized data files are processed which contain information about resources defined or running on a system. Often times these discovery processes cannot determine which system a resource is running on or where it is located within the enterprise. That is where aggregation comes in. Aggregation is the concept of grouping or categorizing objects within Tivoli Business Systems Manager (TBSM) based on the object's attribute values. This is especially useful during the Discovery process where hundreds or thousands of objects may be discovered and inserted into the TBSM database and their physical whereabouts are unknown.

TBSM uses aggregation in its deployment of the management of CICSplex objects. This is because the CICSplex discovery process cannot guarantee knowledge of where every CICSplex resource is running. When resources are inactive, they are not running and therefore that information is unavailable. In lieu of discarding an inactive discovered resource, the resource is represented in an aggregation folder within the Enterprise Outliner. In the case where CICSplex resources are discovered while active, the object is represented in two places in the Business Object Container (BUSC), the aggregation folder and the

Operating System (OS) that the resource is running on. Although the object appears twice in the BUSC it is a single object with a single unique object id.

Aggregation Rules

Aggregation is performed by a series of rules that are managed and maintained from the CICSplex object within TBSM. The CICSplex object contains a tab on the Property Sheet that lists all of the active rules as well as provides controls for performing various operations on the rules. The operations available are the following:

Create a New Aggregation Rule

- Opens the Create Aggregation Rule window.

Delete the Currently Selected Aggregation Rule

- Deletes the currently selected aggregation rule in the list view on the aggregation tab.

Edit the Currently Selected Aggregation Rule

- Opens the Aggregation Rule Properties window.

Refresh the List of Aggregation Rules

- Refreshes the list of aggregation rules on the Aggregation Tab with all of the rules contained in the database.

Reapply all of the Aggregation Rules in the CICSplex

- Re-applies any rule changes to the database. This should be run after any rule modification occurs.

Aggregation Rules are created and edited in the Aggregation Rule Properties window. This window provides fields for the name, description, folder name and priority of the rule. The Name field is the name of the Rule itself, and the Description is a text field where a brief description of the rule can be added by the author. The folder name is the name of the Aggregation Folder that contains the objects that meet the evaluation criteria of the rule. The Priority field of the rule allows the author to prioritize the rule in relative relationship to the other rules that are enabled. The higher the number the higher the priority. For example, a rule with a priority of 5 is evaluated prior to a rule with a priority of 3. The default priority of all rules is 0. The Enabled checkbox flags the rule for evaluation. If checked the rule is evaluated during a CICSplex discovery. If the Enabled checkbox is not checked, the rule is disabled and consequently is not evaluated.

The Exclusive checkbox determines whether the rule is exclusive or not. During Discovery processing, all rules are potentially evaluated for each object. An exclusive rule evaluates objects from the discovery and if the criteria are met for the object, the object is placed in the aggregate folder that the rule designates and no other rules are evaluated for that object. If the criteria are not met for the exclusive rule, the next rule in the chain is evaluated. If the rule is non-exclusive, and the criteria are met, the object is placed in the aggregate folder that the rule designates and the next rule in the chain is evaluated. Non-exclusive rules have the potential of more than one rule being satisfied which places the object in multiple Aggregation Folders.

In the Aggregation Rule Properties window, the Pattern Matching fields and the MAS Types list are the filtering criteria for the evaluation of MASs or CICS Regions. The MAS Types is a list of attribute values to define the following:

Unknown

Discovery could not collect the MAS type attribute from the MAS or CICS Region.

Unmanaged

The MAS or CICS region is not managed by CICSplex

Local The MAS or CICS Region is running on the same image as its CMAS (Management Region).

Remote

The MAS or CICS Region is running remotely from its CMAS (Management Region).

The Pattern Matching fields allow you to create rules that can use pattern matching to evaluate objects based on the MAS Name, Description or System that the MAS is running on. The rules for pattern matching follow the SQL standard whereby the '%' (percent) symbol is used to indicate any number of characters or the '_' (underscore) symbol is used to indicate a single character wildcard. Any characters between '[' and ']' will be treated as a single character that matches the values specified between the brackets. For example '[AaBbCc]' indicates a match for any single character that is an upper or lower case a,b, or c. Ranges of characters may also be specified this way. For example, '[A-Z0-9]' indicates any upper-case character in the range of A through Z, or any numerical digit between 0 through 9. If the '^' (carat) symbol appears immediately following the '[' this indicates that the remaining characters should not match a character in the data. For example, '[^A-Z]' specifies a single character that is not in the range of A through Z.

6

Event Processing Scenarios

Information in this chapter describes two event processing scenarios. The first scenario is a CMAS reporting a short on storage state, and the second scenario is CMAS reporting that a short on storage state has been corrected.

CMAS Reports a Short on Storage State

The sequence of events for CMAS reporting a short on storage state is as follows:

1. A CMAS enters a Short on Storage (SOS) state.
2. CICSplex SM detects change and generates generic SNA alert.
3. NetView traps the message through the MAT.
4. The status is passed to the Tivoli Business Systems Manager where the status for the object that represents the CMAS resource is modified.

CMAS Reports Short a Storage State has been Corrected

The sequence of events for CMAS reporting that a short on storage state has been corrected is as follows:

1. A CMAS reports that the Short on Storage (SOS)) state has been corrected.
2. CICSplex SM detects a return to normal state and notifies NetView via generic SNA alert.
3. NetView traps the message through the MAT.
4. The status is passed to the Tivoli Business Systems Manager where the status for the object that represents the CMAS resource is modified.

7

Using CICSplex SM Instrumentation Tasks

Available Tasks

This chapter describes available instrumentation tasks and explains how to access these tasks from TBSM.

Tasks available for the CICS regions and what they do are:

Query State

Returns the current state of the component. This can be a CMAS or CICS component.

Show ACB

Displays the current value of the VTAM ACB status for the CICS component.

Show IRC

Displays the current value of the InterRegion Communication (IRC) logon status for the CICS component.

To access these tasks and execute the commands from the TBSM Tree View window:

1. Select a CICS Region and then click the right mouse button on the region to open a menu.
2. On the menu, select **Commands**.
3. Select one of the available tasks (**Query State**, **Show ACB**, or **Show IRC**).
4. When the results are captured and presented, click the **Execute** button to execute the command.

8

Troubleshooting

If it is observed that the Windows NT based components of TBSM are not receiving the instrumentation event data from CICSplex SM, then it is necessary for the system programmer or TBSM administrator to perform problem determination. The correct data flow from the origin of a problem condition in CICS to TBSM is as follows:

1. An abnormal condition occurs in a subsystem such as CICS.
2. The CICSplex SM monitor detects the condition and generates an alert or message, which is transmitted to Tivoli NetView for OS/390.
3. The Tivoli NetView for OS/390 Application Management Interface (AMI) determines the processing routine associated with this alert or message.
4. The processing routine formats the alert/message and sends it to the TBSM Source/390 Object Pump via the Program to Program Interface (PPI).
5. The TBSM Source/390 Object Pump queues the alert/message in the data space.
6. The TBSM Source/390 Object Server dequeues the alert/message and forwards it to the TBSM via the LU6.2 session.

Diagnostic Activities

If the flow is interrupted, the system programmer or TBSM administrator should diagnose the failure point and then attempt to fix the problem. Since the flow involves several tasks, each task must be checked individually. The following list displays the diagnostic tasks that should be checked to ensure proper operation.

- Verify that CICS and CICSplex Management Address Spaces (CMAS) are running.
- Identify the Jobname/Taskname for the CICS region and the CMAS region. Use the SDSF Display Active command or display the active tasks from the operator console.
- Verify that the PPI Receiver NETVAOP is active from Tivoli NetView for OS/390 perspective. This informs you if the PPI connection between Tivoli NetView for OS/390 and TBSM Source/390 Object Pump is functional.
- Issue the DISPPI command from a Tivoli NetView for OS/390 operator console. The list of receiver names and buffer statistics is displayed. NETVAOP receiver should be listed.
- Issue the AUTOTBL STATUS command from a Tivoli NetView for OS/390 operator console. Verify that the automation table containing the instrumentation code supporting the CICSplex SM is active (IHSCMAT1).
- If tasks fail to respond or execute properly, verify the NETCONV connection exists between Tivoli NetView for OS/390 and the TBSM server. Check the Tivoli NetView for OS/390 log for errors.

Problem Determination

This section discusses problems and appropriate actions to remedy them.

- **Problem:** The PPI Receiver name NETVAOP is not listed in NetView.

Actions:

- Check the PPI status from the AOP.
- Verify the NetView SSI subsystem is running with the PPI option enabled.

F AOP,PPI STATUS

- **Problem:** The PPI Receiver name NETVAOP is listed but the automation table is not functioning.

Actions:

- The system programmer responsible for NetView needs to verify that the AMI support has been installed correctly. The automation table containing the AMI statements should be checked that it is active.
- Verify the PPI Receiver NETVAOP is active from AOP's perspective by issuing the following command to display the status of the PPI Receiver.

Issue AUTOTBL STATUS and ensure IHSCMAT1 is active

- **Problem:** The PPI Receiver task is not running within AOP or the PPI is disabled.

Actions:

- Review the AOP job log and determine the state of the PPI. If there are no messages indicating the state of the PPI issue the following command.

F AOP,PPI ENABLE

- If the PPI command is not accepted, GTM7837E PPI OPTION IS INVALID then the PPI task is not running within AOP's address space.

- Verify that PPI=YES is specified in the object pump's startup parameters.

If the PPI is enabled successfully then the PPI Receiver was inactivated by a PPI DISABLE command

- Verify data is received in the object server LOG Files.

Issue the F OBJSRVR,LOGSWITCH command to switch the log files. Review the job log of object server and determine the log file that is inactive. This is the file that was used prior to the LOGSWITCH command.

- **Problem:** Object Server LOG Files are not receiving data.

Actions:

- Browse the inactive log file and determine if any records were written to that file. Assuming there has been no activity, verify the status of the PPI Receiver in NetView and AOP.

- Verify if any records were received on the TBSM DAT FILE

The TBSM DAT FILES are located on the machine that runs the SNA Client ASIMVSListenerSvc program and event handler services. The DAT files can be found in the following directory.

drive:\TivoliManager\Data\smfid_date_timestamp.dat

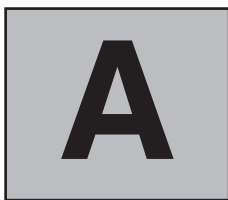
The same records that are viewed in the object server LOG FILE should also be viewed in the TBSM DAT FILE.

-
- **Problem:** The active DAT file is not receiving the AMI data from TBSM Source/390 Object Server.

Actions:

- Verify that TBSM Source/390 Object Server is receiving data from the PPI.
- Verify the status of the SNA communication environment supporting the LU6.2 pipe. Check the SNA Server, SNA Client, and VTAM environments to determine the state of all of these machines.

Correct the SNA connectivity problems. Restart TBSM Source/390 Object Server.



CICSplex Class Definitions

The following table lists the new classes supported for CICSplex monitoring. Files and Transactions have been included even though they have been supported for several releases. This is because CICSplex monitoring generate events that reflect the availability of both of these resources.

Object Type	Description
CICSplex	The highest level object within the CICSplex SM data model. CICSplex SM treats a whole collection of CICS regions as an integrated whole. It consolidates these regions into one or more CICSplexes and lets you manage the CICSplex as a single CICS region image.
CMAS	CICS Managing Address Space (CMAS). The hub of any CICSplex configuration. Provides most of the CICSplex SM functions including the single-system image. It provides services such as message, trace, program call, locking, and event notification for event driven management of the CICSplex.
CICS	CICS Managed Address Space. CICS regions being managed by the CICSplex SM. MASs are defined and managed by a part of a CICSplex. Each MAS in a CICSplex is managed by a CMAS. MASs in a CICSplex can be managed by several CMASs, but only one CMAS is defined as the maintenance point. A MAS can be local or remote, depending on the communication link with the CMAS.
File	A dataset under CICS
Transaction	A series of processing steps that are treated as a single activity to perform a desired result.

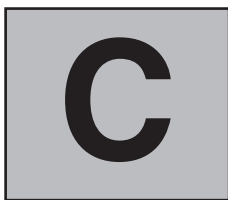
B

CICSplex Class Attributes

For each of the CICSplex SM classes Tivoli Business Systems Manager maintains the following attributes:

Object Type	Attribute	Description	Type	Values
CICSplex				
	CICSplexSMName	CICSplex SM Name	Text(48)	
	CICSplex SM Desc	CICSplex SM Description	Text(64)	
CMAS				
	CICSplex SMName	CMAS Name	Text(48)	
	Desc	CMAS description	Text(54)	
MAS				
	MASName	MAS Name	Text(20)	
	MASType	MAS type		LOCAL,REMOTE
	CICSplex Name	Name of CICSplex belonging to	Text(8)	
	CMAS Link	Name of CICSplex SM connected to	Text(8)	
	MAS Applid	VTAM Appl ID of MAS	Text(8)	
	MASDesc	MAS Description	Text(64)	
Files				
	MASName	MAS Name	Text(20)	
	MASType	MAS type	Text(8)	LOCAL,REMOTE
	CICSplex Name	Name of CICSplex belonging to	Text(8)	
	File Name	File Name Used by CICS	Text(8)	
	File Status	File Status		Enabled,Disabled,Unenabled,Enabling
	File Status	File Status		Open/Closed
Transaction				
	MASName	MAS Name	Text(20)	

Object Type	Attribute	Description	Type	Values
	MASType	MAS type	Text(8)	LOCAL,REMOTE
	CICSplex Name	Name of CICSplex belonging to	Text(8)	
	MASName	MAS Name	Text(20)	
	Transaction Name		Text(4)	
	Transaction Status	Transaction Status	Text	Enabled/Disabled



CICSplex Class States

Registered objects within TBSM have the following states:

Object Type	States
CICSplex	
CMAS	Active, Inactive
CICS	Active, Inactive
File	Open/Closed, Enabled/Disabled, Unenabled/Enabling
Transaction	Enabled, Disabled



IHSCCPPD Messages

IHSCCPPD writes a number of messages to the SYSPRINT DD dataset. Some are informational, and some are error messages, which result in the stopping of IHSCCPPD.

IHSCCPPD CICSplex Name for this run is XXXXXXXX

Explanation: The CICSplex Name entered in the PARM field of the EXEC IHSCCPPD statement is XXXXXXXX. If this is not a valid CICSplex name, IHSCCPPD stops.

IHSCCPPD CMAS Name for this run is XXXXXXXX

Explanation: The CMAS Name entered in the PARM field of the EXEC IHSCCPPD statement is XXXXXXXX. If this is not a valid CMAS name, IHSCCPPD stops.

IHSCCPPD Security value for this run is XXXXXXXX

Explanation: The Security Value entered in the PARM field of the EXEC IHSCCPPD statement is XXXXXXXX. This value is optional but, if required, an incorrect value could cause a failure on CONNECT.

IHSCCPPD CPSM Release for this run is XXXX

Explanation: The Release entered in the PARM field of the EXEC IHSCCPPD statement is XXXXXXXX. This value is mandatory and an incorrect value could cause a failure on CONNECT.

IHSCCPPD UserID for this run is XXXXXXXX

Explanation: The UserID entered in the PARM field of the EXEC IHSCCPPD statement is XXXXXXXX. This value is optional but, if required, an incorrect value could cause a failure on CONNECT.

IHSCCPPD Total number of CMAS objects is nnnnn

Explanation: The total number of CMAS objects discovered for this CICSplex is nnnnn.

IHSCCPPD Total number of active CMAS objects is nnnnn

Explanation: The total number of active CMAS objects discovered for this CICSplex is nnnnn.

IHSCCPPD Total number of MAS objects is nnnnn

Explanation: The total number of MAS objects discovered for this CICSplex is nnnnn.

IHSCCPPD Total number of active MAS objects is nnnnn

Explanation: The total number of active MAS objects discovered for this CICSplex is nnnnn.

IHSCCPPD Processing records for MAS XXXXXXXXX

Explanation: IHSCCPPD is processing detail (transaction and file) records for MAS XXXXXXXXX. This message is issued for each active MAS discovered.

IHSCCPPD CPSM Command GET OBJECT(XXXXXXX) Failed, Response code = nnnnn Reason code = mmmmm

Explanation: The CICSplex SM GET for object XXXXXXXX failed. In some cases, this is not an error, for example when an active MAS has no remote files the following message would be issued:

IHSCCPPD CPSM Command GET OBJECT(REMFILE) Failed, Response code = 01027 Reason=

IHSCCPPD terminates if a GET fails for either a CMASLIST object (getting the list of all known CMAS objects) or a MAS object (getting the list of all known MAS objects). For any other GET failure, processing continues with the next object.

IHSCCPPD CPSM Command QUERY Failed, Response code = nnnnn Reason code = mmmmm

Explanation: The CICSplex SM QUERY to obtain information from the object retrieved by the previous GET has failed. IHSCCPPD will terminate. The Response and Reason codes are documented in SC33-1430 IBM CICSplex System Manager for MVS/ESA Application Programmer Interface.

IHSCCPPD CPSM Command FETCH(XXXXXXX) Failed, Response code = nnnnn Reason code = mmmmm

Explanation: The CICSplex SM FETCH to obtain information from the object type XXXXXXXX has failed. IHSCCPPD will terminate. The Response and Reason codes are documented in SC33-1430 IBM CICSplex System Manager for MVS/ESA Application Programmer Interface.

IHSCCPPD CPSM Command DISCARD(XXXXXXX) Failed, Response code = nnnnn Reason code = mmmmm

Explanation: The CICSplex SM DISCARD to release the object XXXXXXXX retrieved by the previous GET has failed. IHSCCPPD terminates. The Response and Reason codes are documented in SC33-1430 IBM CICSplex System Manager for MVS/ESA Application Programmer Interface.

IHSCCPPD Total number of records is nnnnnnn

Explanation: The total number of records produced by this execution is nnnnnnn.



Creating REXX Function Package Aliases

Use the following sample as a guideline if you want to create IRXFLOC and DSIRXLFP REXX function package aliases. The data set used in the SYSLMOD data definition (DD) statement must be authorized for the authorized program facility (advanced printer function (APF)) and concatenated to the STEPLIB DD statement in the procedure used to start Tivoli NetView for OS/390. We recommend that this data set be the first data set in the concatenation stream.

```
//LKEDJOB MSGLEVEL=(1,1),REGION=4M

//*****

//*          LINKEDIT OF EYU9AR00 TO ADD ALIAS OF DSIRXLFP

//*****

//EXEC PGM=IEWL,REGION=4096K,

//PARM='RENT,REUS,NCAL,LIST,MAP,LET,

//AMODE=31,RMODE=ANY

//SYSLIBDD DISP=SHR,DSN=CPSM130.SEYAUTH

//SYSLMOD      DD DISP=SHR,DSN=CPSM.USERLINK

//SYSUT1DD UNIT=SYSDA,SPACE=(TRK,(30,30))

//SYSPRINTDD SYSOUT=*

//SYSLINDD *

INCLUDE SYSLIB(EYUPAR00)

ORDER      EYU9AR00

ENTRY      EYU9AR00

MODE       AMODE(31)

MODE       RMODE(ANY)

ALIAS      EYU9AR01
```

ALIAS	IRXFLOC
ALIAS	DSIRXFLP
SETCODE	AC(1)
NAME	EYU9AR00(R)



AOP Input Parameters

AOP Initialization parameters are input via the AOPSYSIN DD card (LRECL=80, RECFM=F or FB). Most are optional and defaults are used if not supplied. Some, however, are required and some functions of AOP may not be available if these are omitted.

The ACC1IDxx DD card in the AOP startup JCL supplies a two character ID (the xx characters of the DD name) used to identify and connect each set of TBSM Source/390 Object Pump, Dataspace, and Server regions (the Tivoli Business Systems Manager regions) on an MVS system.

The characters default to 01 if no ACC1IDxx DD card is included in the startup JCL. By using a different suffix for each set of TBSM regions, you can run multiple copies of Tivoli Business Systems Manager on a single MVS system (for example for testing purposes).

You can also cause AOP to insert the ACC1IDxx characters into AOP startup parameters, thus allowing you to automatically configure AOP depending upon the ID characters in use. To do this, code two dots (..) at each point in the parameter where you want the characters to be inserted. For example, each AOP region on an MVS system needs its own MCS console name. By coding .. in the `CONSOLE_NAME` parameter operand, AOP inserts the ID chars into the name, thus ensuring that the name is unique within that MVS system.

Thus if:

```
CONSOLE_NAME=ACCPRD..
```

and the ACC1IDxx DD card is coded as:

```
//ACC1IDPD DD DUMMY
```

the console name used by AOP will be ACCPRDPD

Continuing Parameters

If a parameter needs to be continued onto one or more cards then specify a comma followed by a space at the end of the data and start the continued data in column one of the next card.

Thus:

```
KEYWORD=partof,  
thedata
```

is the same as:

KEYWORD=partofthedata

Notice that the comma is NOT part of the data for the keyword. If you want a comma to be part of the data code two commas. Either of the following is valid:

KEYWORD=part1,,
part2

or

KEYWORD=part1,',
'part2'

Both of these are the same as

KEYWORD=part1,part2

The data for KEYWORD will be "part1,part2". The first comma is included in the data, the last comma indicates that the data is continued on the next card.

Parameters

INITIAL_EXEC=@AOPEXEC | name

Specifies the one to eight character name of the initial REXX EXEC that AOP runs, once initialization has completed. The default is @AOPEXEC. You can change this to run a different initialization REXX EXEC if required.

CONSOLE=TM390.. | name | NO

Specifies the one to eight character name to be assigned to the AOP MCS console. The default is TM39001 (where .. are the ID chars from the ACC1ID DD card in the AOP startup JCL, or 01 if omitted).

If CONSOLE=NO is coded then console messages are not trapped by AOP and operator commands cannot be issued (by the REXX OPER command, or the OPER action in traps).

CONSOLE_MSCOPE=LOCAL | ALL

Specifies the message scope of the console. LOCAL specifies that the console only receive messages from the local system (the one AOP is running on). ALL specifies that the console will receive messages from all systems in the Sysplex.

The default is LOCAL.

MAX_TRAPS = 2016 | number

Specifies the maximum number of traps that AOP is able to define. The actual number that can be defined may be slightly smaller than this, depending upon the mix of trap types defined. The number defined will be rounded UP to a multiple of 32. If storage is not available for the defined number of traps then AOP calculates the maximum number of traps that it can handle in the storage available. The

number can be in the range 512-20000.

TERMINAL_PREFIX=prefix | TM390

Specifies a one to seven character prefix that is used as a prefix to build the LU names of the terminals used to logon to Omegamon and other VTAM applications.

If no prefix is supplied VTAM services will NOT be available in AOP. The default is TM390.

NUMBER_OF_TERMINALS=0 | number

Specifies the number of virtual terminals to be available to AOP in the range 0-99999. If 0 is specified or allowed to default then VTAM services are not available to AOP. A pool of virtual terminals is created by using the prefix value from the TERMINAL_PREFIX operand with a number in the specified range.

The combined length of the prefix and the number of digits in the largest number must not exceed 8. Thus if the prefix is AOPLU2, the maximum number of virtual terminals is 99 (giving AOPLU201 through AOPLU299).

The numeric suffix of each terminal is zero filled and right justified to the number of digits in the NUMBER_OF_TERMINALS operand, Thus if NUMBER_OF_TERMINALS=80 (2 digits) then the numeric suffix of each terminal will be 01, 02 etc. to 80. If NUMBER_OF_TERMINALS=200 (3 digits) then the numeric suffix of each terminal will be 001, 002 etc. to 200.

MODEL2_LOGMODE=D4A32782 | name

Specifies the logmode name to be used for model 2 (24 x 80) terminals. This should be an SNA non queriable logmode.

MODEL3_LOGMODE=D4A32783 | name

Specifies the logmode name to be used for model 3 (32 x 80) terminals. This should be an SNA non queriable logmode.

MODEL4_LOGMODE=D4A32784 | name

Specifies the logmode name to be used for model 4 (43 x 80) terminals. This should be an SNA non queriable logmode.

MODEL5_LOGMODE=D4A32785 | name

Specifies the logmode name to be used for model 5 (27 x 132) terminals. This should be an SNA non queriable logmode.

LOG_SCREEN=NO | YES

Specifies whether screen images from the VTAM 3270 virtual session should be logged or not.

The default is NO.

EDI=NO | YES

Specifies whether AOP is to initialize the External Data Interface.

The default is NO.

EDI_BUFFER_SIZE=2048 | size

Specifies the size (in K, 1K = 1024 bytes) of the External Data Interface buffer.

The default is 2048 K (2 Megabytes).

EDI_BUFFER_SIZE=2048 | size

Specifies the size (in K, 1K = 1024 bytes) of the External Data Interface buffer.

The default is 2048 K (2 Megabytes).

Size can be in the range 50 (50 K) to 20480 (20 Megabytes).

EXEC_REFRESH=YES | NO

Specifies whether REXX EXECs will be loaded from disk each time (YES) the default, or only loaded once and then held in storage (NO).

YES is suitable for a development environment where REXX execs may be being changed while AOP is running.

NO is suitable for a production environment where the performance impact of continually re-loading the REXX EXECs is best avoided.

SUBSYSTEM_NAME=AS.. | name | NO

Specifies the four character name of the subsystem created and used by AOP.

The default is AS.. (where .. are the ID chars from the ACC1ID DD card in the AOP startup JCL, or 01 if omitted).

Coding the value NO will prevent AOP from initializing the AOP subsystem. In that case command (CMD) traps will not be processed as no system commands will be captured.

SUBSYSTEM_MAXTHREADS = 2 | nn

Specifies the maximum number of concurrent operator commands that can be processed by the AOP subsystem/address space. The default is 2.

nn can be in the range 1 to 50.

MAXTHREADS_PROMPT = YES | NO

Specifies whether AOP should prompt the operator to increase the number of subsystem threads or not by issuing message ACC7560 when SUBSYSTEM_MAXTHREADS is reached by the subsystem command interface.

YES indicates that AOP should issue the WTOR message. NO indicates that AOP should not issue the WTOR message.

If NO is coded the operator is NOT prompted to increase the maximum number of subsystem command threads when the limit (SUBSYSTEM_MAXTHREADS) is reached. Some operator commands may not be trapped by AOP in this case.

LOAD=program_name

Causes AOP to preload the named program and save its entry point in an internal table which can then be used at trap match time to CALL a program when a trap match occurs. This parameter can be specified as many times as required in the input parm stream.

CUSE_DATASPACE = YES | NO

Specifies whether the region should connect to the TBSM Source/390Dataspace or not. The default is YES. The TBSM Source/390Dataspace must be initialized and the server running before AOP is started when YES is coded.

CMD=command,PROGRAM=program

Defines a modify command that AOP recognizes and the handling program that is invoked to process the command.

CALL program

ATTACH program [,TYPE=MAIN]

The CALL and ATTACH statements can only be coded within the region initialization table. They cannot be coded as user input in the AOPARM input deck. They define the sequence and names of programs to CALL or ATTACH during AOP startup to initialize the region.

TYPE=MAIN indicates that the attached program runs as an extension of MAIN and is shutdown last after all subtasks have ended. It should NOT be used for most normal managers.

OMEGAMON_REFRESH_PERIOD=hh:mm:ss | 00:04:00

Defines the default setting for how frequently Omegamon sessions should be polled for exceptions. The default setting is 4 minutes.

OMEGAMON_EXCEPTION_CNTRL=ALL | DELTA

Specifies if ALL Omegamon exceptions should be sent to NT (the default) or only changes.

TRAP_REPORT=destination | SYSOUT(A)

Specifies the destination for trap reports. The default is SYSOUT class A.

LOG_FILE=destination | SYSOUT(A)

Specifies the destination of the log file. The default is SYSOUT class A.

LOG_FILE_RECORD_LENGTH=length | 200

Specifies the lrecl for the log file. The default is 200.

LOG_SPIN_INTERVAL=hh:mm:ss | 04:00:00

Specifies how often the log file should be spun off. The default is 4 hours.

COMMAND_PREFIX=string

Specifies the character string that AOP recognizes as commands intended for the pump. This should be unique within your system and not duplicate any existing command string, for example \$ (for JES) or D for system commands.

SHUT_OBJECT_SERVER=YES | NO

Specifies whether the pump shutdowns the TBSM Source/390 Object Server when the pump is terminated. YES indicates that the pump terminates the TBSM Source/390 Object Server. NO indicates that it will not. The default is NO.

SHUTDOWN_DELAY=n | 0

Specifies the number of seconds that the pump waits before terminating. The default is zero seconds. n is the number of seconds to wait in the range 0 to 30 where zero specifies no wait occurs.

OPC_JOBNAME=jobname [,jobname,jobname,...]

Specifies one or more job names for OPC address spaces. A Generalized EDI message trap is created for each jobname specified.

SA390_JOBNAME=jobname [,jobname,jobname,...]

Specifies one or more job names for SA/390 address spaces. A Generalized EDI message trap is created for each jobname specified.

AFOPER_JOBNAME=jobname [,jobname,jobname,...]

Specifies one or more job names for AF/Operator address spaces. A Generalized EDI message trap is created for each jobname specified.

OPSMVS_JOBNAME=jobname [,jobname,jobname,...]

Specifies one or more job names for OPS/MVS address spaces. A Generalized EDI message trap is created for each jobname specified.



TBSM Source/390 Object Pump Modify Commands

The TBSM Source/390 Object Pump accepts the following MVS Modify commands (F *pump,command* [*options*]):

REFRESH

The REFRESH command causes the TBSM Source/390 Object Pump to re-load a REXX EXEC into storage, when running with EXEC_REFRESH=NO (No automatic refresh of REXX EXECs). This can be used if a REXX EXEC has changed and needs to be re re-loaded without restarting the TBSM Source/390 Object Pump. It is not valid if EXEC_REFRESH=YES was coded in the startup parameters, and has no effect if the EXEC being refreshed was not previously loaded.

Syntax: F *pump*,REFRESH *execname*

Where: *Execname* is the name of the REXX EXEC to reload.

LOGSCREENS

The LOGSCREENS command allows you to turn 3270 screen logging on or off dynamically while the TBSM Source/390 Object Pump is running. Logged screens are written to the AOPLOG DD.

Syntax: F *pump*,LOGSCREENS ON | OFF

Where:

- ON specifies that screen logging should be started.
- OFF specifies that screen logging should be stopped.

IF the command is issued without either ON or OFF being specified then the command displays the current status of screen logging.

SHOW

The SHOW command can be used to display the various resources within the TBSM Source/390 Object Pump.

Syntax: F *pump*,SHOW *resource* [*options*]

Where:

- *Resource* I the resource type to display.
- Options are any options valid for the resource type.

SHOW TRAPS

The SHOW TRAPS command displays information about the selected traps and/or trap types.

Syntax: F *pump*,SHOW TRAPS *mask type* [LONG | NOLIST]

Where:

- *Mask* is a generic or specific trap name that is used to select the trap name(s) to display. The default is * (all traps of the selected type).
- *Type* is the trap type, which can be one of WTO, TOD, XOM, XOC, XOI, XO2, CMD or *. The default is * (all trap types).
- LONG specifies that the long form of the selected traps be displayed. The long form includes all the information about the trap and the action that it takes. If the short form is allowed to default then only the trap name and the number of matches is displayed for each trap.
- NOLIST specifies that only the summary line is to be displayed, which shows a count by trap type for the selected traps (bask on mask).

SHOW SESSIONS

The SHOW SESSIONS command displays information about the current 3270 sessions.

Syntax: F *pump*,SHOW SESSIONS *mask*

Where: *Mask* is a generic or specific session name that is to be used to select the session name(s) to display. The default is * (all sessions).

SHOW POOLS

The SHOW POOLS command displays the names of the currently defined global variable pools.

Syntax: F *pump*,SHOW POOLS [*mask*]

Where: *Mask* is a generic or specific pool name to select the pools to be displayed. The default is * (all pools).

SHOW VARS

The SHOW VARS command displays the variable names and currently assigned values of the selected variables in the selected pool(s).

Syntax: F *pump*,SHOW VARS *varnamemask poolnamemask*

Where:

- *Varnamemask* specifies a generic or specific variable name to display. The default is * (all variables in the selected pools).
- *Poolnamemask* specifies a generic or specific pool name that is to be used to select the pool(s) to scan for the selected variable names. The default is * (all pools).

Two pools are always defined. They are the following:

- System Pool - Contains Read only variables.
- User pool - The User pool always exists, even when empty. Contains work variables used by the TBSM Source/390 Object Pump.

SHOW MSGCOUNT

The SHOW MSGCOUNT command displays the number of console messages received by the console interface.

Syntax: F *pump*,SHOW MSGCOUNT

SHOW PPICOUNT

The SHOW PPICOUNT command displays the number of application instrumentation (AMI) messages received by the PPI interface.

Syntax: F *pump*,SHOW PPICOUNT [* | *type*]

Where: *Type* specifies the format type for which the count is to be displayed. The following values are valid types:

- DB2
- IMS
- CPSM
- AMI

RODM

The RODM command can be used to start or stop the TBSM Source/390 Object Pump RODM Interface.

Syntax: F *pump*,RODM START | STOP

Where:

- START indicates that the TBSM Source/390 Object Pump should attempt to start communications with RODM
- STOP indicates that the TBSM Source/390 Object Pump should stop communications with RODM.

PPI ENABLE

Starts the TBSM Source/390 Object Pump PPI Interface.

Syntax: F *pump*,PPI ENABLE

PPI DISABLE

Stops the TBSM Source/390 Object Pump PPI Interface.

Syntax: F *pump*,PPI DISABLE

PPI STATUS

Displays the status of the TBSM Source/390 Object Pump PPI receiver task.

PPI TRACE

The PPI TRACE command will start, stop or display the PPI trace.

Syntax: F *pump*,PPI TRACE [ON\YES | NO\OFF]

Where:

- ON (or YES) specifies that PPI tracing should start. Data received by the PPI Listener is written to the TBSM Source/390 Object Pump AOPLOG DD.
- OFF (or NO) specifies that tracing by the PPI Listener should stop.

If no option is specified the command displays the current state of PPI tracing.

PPI RESET

Displays or resets the PI buffer queue limit.

Syntax: F *pump*,PPI RESET [*buffer_queue_limit*]

Where: *Buffer_queue_limit* specifies the new limit to be applied to the PPI buffer. The valid range is 100 to 10000. If the *buffer_queue_limit* operand is omitted then the command displays the current setting for the buffer queue limit.

If no option is specified the command displays the current state of PPI tracing.



Using the NetView IHSCGLBL Command

The NetView IHSCGLBL Command provides update and listing capability for the NetView global variables that are used for the CICSplex SM SubName capability. This command supports three functions:

- Adding or updating a relationship between the CICSplex Name and a SubName
- Deleting a relationship between a CICSplex Name and a SubName
- Listing the SubName related to a CICSplex Name.

This function must be run on the NetView which resides on the same MVS image as the reporting CMAS for the CICSplex.

To add or update a SubName for a CICSplex, enter the following command on a NetView console:

IHSCGLBL CICSplexName=cccccccc SubName=ssssssss

Where cccccccc is the CICSplex Name as defined to CICSplex SM, and ssssssss is the new SubName to be associated with that CICSplex.

To delete the relationship between the CICSplex Name and a SubName, enter the following command on a NetView console:

IHSCGLBL CICSplexName=cccccccc DELETE=YES

Where cccccccc is the CICSplex Name as defined to CICSplex SM.

To list the SubName related to a CICSplex Name, enter the following command on a NetView console:

IHSCGLBL CICSplexName=cccccccc LIST=YES

Where cccccccc is the CICSplex Name as defined to CICSplex SM.



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