Tivoli® System Automation for z/OS

IBM

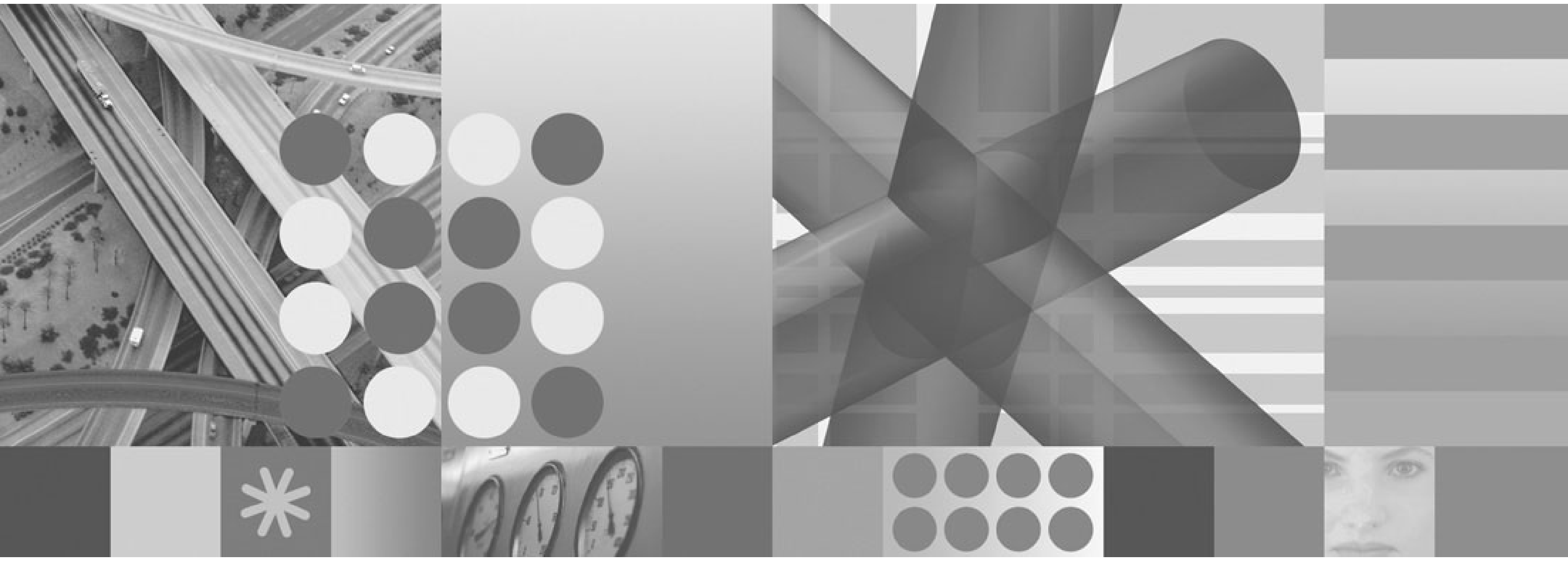
Version 3 Release 2

# Planning and Installation

SC33-8261-04

Tivoli® System Automation for z/OS

Version 3 Release 2



# Planning and Installation

SC33-8261-04

**Note!**

Before using this information and the product it supports, read the information in "Notices" on page xiii.

This edition applies to IBM Tivoli System Automation for z/OS (Program Number 5698-SA3) Version 3 Release 2, an IBM licensed program, and to all subsequent releases and modifications until otherwise indicated in new editions.

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

IBM welcomes your comments. A form for readers' comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

If you prefer to send comments electronically, use one of the following methods:

FAX (Germany): 07031 + 16-3456
FAX (Other Countries): (+49)+7031-16-3456
Internet: s390id@de.ibm.com

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

# Contents

## Chapter 4. Planning to Install TEC Notification by SA z/OS . . . . . . . 45

## Chapter 5. Planning for the NMC Environment . . . . . . . . . . . . 49

## Chapter 6. Planning for Automation Connectivity . . . . . . . . . . . . 53

## Chapter 7. Naming Conventions . . . . 63

# Part 2. Installation . . . . . . . . . . 69

## Chapter 8. Installing SA z/OS on Host Systems . . . . . . . . . . . . . . . 71

# Figures

# Tables

# Notices

This information was developed for products and services offered in the U.S.A.

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

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

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

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

## Programming Interface Information

This publication documents information that is *not* intended to be used as a Programming Interface of System Automation for z/OS.

## Trademarks

The following terms are trademarks or service marks of the IBM Corporation in the United States or other countries or both:

AIX
CICS
DB2
DFS
DFSMS/MVS
ESCON
GDPS
IBM
IMS
MQSeries
Multiprise
MVS
MVS/ESA
MVS/SP
MVS/XA
NetView
OS/390
Parallel Sysplex
Processor Resource/Systems Manager
PR/SM
RACF
RMF
S/390
SecureWay
Sysplex Timer
Tivoli
Tivoli Enterprise Console
VM/ESA
VSE/ESA
VTAM
WebSphere
z9
z/OS
z/VM
zSeries

The following terms are trademarks of other companies:

- Java is a trademark of Sun Microsystems, Inc. in the United States and other countries.
- Linux is a trademark of Linus Torvalds.
- Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation and other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.

# Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS™ enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

## Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

## Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

## z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at:

`http://www.ibm.com/servers/eserver/zseries/zos/bkserv/`

# About This Book

This book describes IBM® Tivoli® System Automation for z/OS (SA z/OS) from a planning point of view, and how to install the product.

It also describes how to migrate to the latest release of SA z/OS.

## Who Should Use This Book

This information is intended primarily for system programmers and automation programmers who plan for systems management and who install this product.

## Notes on Terminology

**MVS:**

References in this book to "MVS™" refer either to the MVS/ESA™ product or to the MVS element of z/OS.

**NetView:**

The term *NetView®* used in this documentation stands for *IBM Tivoli NetView for z/OS*.

## Where to Find More Information

### The System Automation for z/OS Library

The following table shows the information units in the System Automation for z/OS library:

*Table 1. System Automation for z/OS Library*

| Title | Order Number |
|---|---|
| *IBM Tivoli System Automation for z/OS Planning and Installation* | SC33-8261 |
| *IBM Tivoli System Automation for z/OS Customizing and Programming* | SC33-8260 |
| *IBM Tivoli System Automation for z/OS Defining Automation Policy* | SC33-8262 |
| *IBM Tivoli System Automation for z/OS User's Guide* | SC33-8263 |
| *IBM Tivoli System Automation for z/OS Messages and Codes* | SC33-8264 |
| *IBM Tivoli System Automation for z/OS Operator's Commands* | SC33-8265 |
| *IBM Tivoli System Automation for z/OS Programmer's Reference* | SC33-8266 |
| *IBM Tivoli System Automation for z/OS CICS Automation Programmer's Reference and Operator's Guide* | SC33-8267 |
| *IBM Tivoli System Automation for z/OS IMS Automation Programmer's Reference and Operator's Guide* | SC33-8268 |
| *IBM Tivoli System Automation for z/OS TWS Automation Programmer's Reference and Operator's Guide* | SC23-8269 |
| *IBM Tivoli System Automation for z/OS End-to-End Automation Adapter* | SC33-8271 |

*Table 1. System Automation for z/OS Library (continued)*

| Title | Order Number |
|---|---|
| *IBM Tivoli System Automation for z/OS Monitoring Agent Configuration and User's Guide* | SC33-8337 |

The System Automation for z/OS books are also available on CD-ROM as part of the following collection kit:

IBM Online Library z/OS Software Products Collection (SK3T-4270)

**SA z/OS Home Page**

For the latest news on SA z/OS, visit the SA z/OS home page at http://www.ibm.com/servers/eserver/zseries/software/sa

## Related Product Information

You can find books in related product libraries that may be useful for support of the SA z/OS base program by visiting the z/OS Internet Library at http://www.ibm.com/servers/eserver/zseries/zos/bkserv/

## Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most of the IBM messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can use LookAt from these locations to find IBM message explanations for z/OS elements and features, z/VM®, VSE/ESA™, and Clusters for AIX® and Linux™:

- The Internet. You can access IBM message explanations directly from the LookAt Web site at http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/.
- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations using LookAt from a TSO/E command line (for example: TSO/E prompt, ISPF, or z/OS UNIX® System Services).
- Your Microsoft® Windows® workstation. You can install LookAt directly from the *z/OS Collection* (SK3T-4269) or the *z/OS and Software Products DVD Collection* (SK3T4271) and use it from the resulting Windows graphical user interface (GUI). The command prompt (also known as the DOS > command line) version can still be used from the directory in which you install the Windows version of LookAt.
- Your wireless handheld device. You can use the LookAt Mobile Edition from http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookatm.html with a handheld device that has wireless access and an Internet browser (for example: Internet Explorer for Pocket PCs, Blazer or Eudora for Palm OS, or Opera for Linux handheld devices).

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from:

- A CD-ROM in the *z/OS Collection* (SK3T-4269).
- The *z/OS and Software Products DVD Collection* (SK3T4271).

- The LookAt Web site (click **Download** and then select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

# Part 1. Planning

This part provides details on the following:

- Chapter 1, “SA z/OS Prerequisites and Supported Equipment,” on page 3
- Chapter 2, “What Is New in SA z/OS 3.2,” on page 9
- Chapter 3, “Planning to Install SA z/OS on Host Systems,” on page 23
- Chapter 4, “Planning to Install TEC Notification by SA z/OS,” on page 45
- Chapter 5, “Planning for the NMC Environment,” on page 49
- Chapter 6, “Planning for Automation Connectivity,” on page 53
- Chapter 7, “Naming Conventions,” on page 63

# Chapter 1. SA z/OS Prerequisites and Supported Equipment

## SA z/OS Components

SA z/OS consists of the following three components:

- System operations (*SysOps* for short)
- Processor operations (*ProcOps* for short)
- I/O operations (*I/O Ops* for short)

Refer to “Component Description” on page 23 for details.

SA z/OS also provides special automation facilities for the following products:

- CICS®
- DB2®
- IMS™
- TWS

## Hardware Requirements

IBM has tested SA z/OS on IBM processors. SA z/OS uses the S/390® interfaces that vendors of other processors capable of running z/OS have stated that they support. Check with your vendor for details.

The target system can run in any hardware environment that supports the required software.

### SA z/OS Processor Operations

The processor operations base program can run on any processor supported by Tivoli NetView for z/OS, Version 5 Release 1.

### SA z/OS System Operations

The system operations base program can run on any processor supported by Tivoli NetView for z/OS, V5.1 and z/OS Version 1 Release 7.

### SA z/OS I/O Operations

The I/O operations base program can run on any processor supported by z/OS V1.7.

**Note:** If an ESCON® channel has not been installed and defined, I/O operations recovers from an ABEND 0C1, issues message IHVD014E, and terminates startup.

## Workstation Components

The NMC exploitation used by SA z/OS can run on all NMC topology server and NMC topology client hardware that is supported by Tivoli NetView for z/OS, V5.1.

## Functional Prerequisites

The hardware interface functions used by the INGPLEX command and the IXC102A message automation without processor operations is supported by the following processor hardware families:

- System z9™
- zSeries®
- CMOS-S/390 G6
- CMOS-S/390 G5

For current information about the LIC levels that are required for these servers, refer to the PSP bucket.

The following processor hardware can be controlled as a target with the BCP internal interface of the above listed processors, but cannot use the SA z/OS BCP internal interface to control itself or other processors:

- CMOS-S/390 G4
- CMOS-S/390 G3

The following micro code levels must be applied to all HMCs and SEs:

| Processor Hardware | Micro Code Levels |
|---|---|
| CMOS-S/390 G3, G4 | Driver A2 F10980.083 |
| CMOS-S/390 G5, G6 | Driver 26 F99918.152 |
| zSeries z800, z900 | Driver 3G J11213.154<br>Driver 3C J10638.116 |
| zSeries z990 | Driver 52 J12560.090 |

These MCL levels are required for all HMCs that serve as Master HMCs and have the LIC change console service enabled. Note that at least one HMC in your processor LAN configuration must have this service enabled in order to provide cross-CPC communication over the BCP internal interface.

### Prerequisites for z9 Processors

The following tables list the Support Element and HMC Micro Code Levels that must be applied to the z9 SEs and HMCs. Both SA z/OS Processor Operations SNMP support and the SA z/OS BCP Internal Interface require these micro code levels.

The following micro code levels must be applied to all z9 SEs:

| Processor Hardware | Driver | Micro Code Level | Bundle |
|---|---|---|---|
| z9 | 63 | J99677.067<br>J99677.097<br>J99677.111<br>J99677.123<br>J99677.128<br>J99677.129<br>J99677.151<br>J99677.152<br>J99677.164<br>J99677.179<br>J99677.187<br>J99677.200<br>J99677.213<br>J99677.219<br>J99678.001 | 7<br>9<br>10<br>10<br>10<br>10<br>12<br>12<br>13<br>13<br>13<br>13<br>14<br>14<br>12 |

The following micro code levels must be applied to all z9 HMCs used for SA z/OS ProcOps or act as a master console for the BCP Internal Interface:

| Processor Hardware | Driver | Micro Code Level | Bundle |
|---|---|---|---|
| z9 | 631 | G34191.084<br>G34191.097<br>G34191.107<br>G34191.114<br>G34191.124<br>G34191.133<br>G34191.136 | 08A<br>09<br>09<br>09<br>10<br>10<br>10 |

**Note:** A number of hardware commands are not supported when running on a z/OS image that runs under z/VM. Refer to*IBM Tivoli System Automation for z/OS Customizing and Programming* for information about which particular functions are affected.

## Software Requirements

This section describes the environment of the target system required to install and use SA z/OS.

**Notes:**

1. To properly invoke the Japanese language version of SA z/OS, a Japanese language version of NetView must be installed and the Kanji support must be enabled. For Kanji workstation support a Japanese language host must be connected to a Japanese language workstation. If an English language workstation is connected to a Japanese language host some messages may be unreadable.
2. Check with IBM Service for required product service levels in addition to the base product releases. Certain service levels may be required for particular product functions.
3. SA z/OS processor operations is enabled on a focal-point system, from which it monitors and controls SA z/OS processor operations target systems. The SA z/OS processor operations target system may also have SA z/OS installed

for its system operations and I/O operations but the processor operations will not be enabled. This section does not describe the SA z/OS Processor Operations target system.

Unless otherwise noted, subsequent versions or releases of products can be substituted.

## Mandatory Prerequisites

A mandatory prerequisite is defined as a product that is required without exception; this product either *will not install* or *will not function* unless this requirement is met. This includes products that are specified as REQs or PREs.

*Table 2. Mandatory Prerequisites*

| Product Name and Minimum VRM/Service Level |
|---|
| z/OS V1.7 or later |
| Tivoli NetView for z/OS, V5.1 or later |

## Functional Prerequisites

A functional prerequisite is defined as a product that is *not* required for the successful installation of this product or for the basic function of the product, but *is* needed at run time for a specific function of this product to work. This includes products that are specified as IF REQs.

*Table 3. Functional Prerequisites*

| Product Name and Minimum VRM/Service Level | Function |
|---|---|
| **z/OS base elements or optional features:** | |
| z/OS SecureWay® Security Server (including RACF® and DCE Security Server components) | For sysplex-based authorization and RACF-based NetView authorization. |
| **Other program products:** | |
| WebSphere® MQ for z/OS, V5.3.1 or later | For sysplex automation and for communication between the automation manager and the automation agents. |
| HTML Browser | For customization reports. To view the HTML file with an Internet Browser, either Microsoft Internet Explorer 5.50 or above, or Netscape 4.72 or above. |
| z/VM 4.3 or later | For VM Second Level Systems support. |
| PTF UA31443 on z/OS V1.7 or z/OS V1.8 or later with z/OS XML System Services | OMEGAMON XE Support |
| IBM Tivoli OMEGAMON II® for MVS V5.2<br>IBM Tivoli OMEGAMON II for CICS V5.2<br>IBM Tivoli OMEGAMON II for IMS V5.1<br>IBM Tivoli OMEGAMON II for DB2 V5.4 | For the following commands:<br>• INGMTRAP<br>• INGOMX |
| IBM Tivoli Monitoring Services (ITMS, 5698-A79) | SA z/OS Monitoring Agent for Tivoli Enterprise Portal support (FMID HKAH320, see also *IBM Tivoli System Automation for z/OS Monitoring Agent Configuration and User's Guide*) |
| IBM CICS Transaction Server V2.2 or later | CICSPlex® based monitoring |
| **Workstation Prerequisites:** | |
| Tivoli NetView for z/OS, V5.1 or later MultiSystem Manager | For SA z/OS topology manager functions |

*Table 3. Functional Prerequisites (continued)*

| Product Name and Minimum VRM/Service Level | Function |
|---|---|
| NetView Management Console topology server and client | For the SA z/OS NMC workstation exploitation |
| NetView 3270 Management Console | For the SA z/OS NMC workstation exploitation |

# Supported Hardware

SA z/OS processor operations supports monitoring and control functions for any of the following processors:

- z9, zSeries and 390-CMOS processors.
- All CMOS processors supporting Operations Command Facility (OCF) not part of the above processor families are supported by processor operations with limited functionality.

SA z/OS processor operations also supports logical partitioning of any of those processors.

SA z/OS provides a wide range of I/O configuration information and control functions for various types of hardware other than processors, though it does not require any of them. The hardware can include channels, control units and devices (both ESCON and non-ESCON), ESCON Directors (they are not required), and hardware used for sysplex coordination such as coupling facilities and External Time Reference (ETR) devices.

## Operator Terminals

SA z/OS supports any display supported by Tivoli NetView for z/OS, V5.1. This is required for access to SA z/OS system operations and processor operations functions through NetView.

SA z/OS supports any display supported by ISPF V4.2 or higher. This is required for access to SA z/OS I/O operations functions and the SA z/OS customization dialogs.

## Operating Systems Supported by Processor Operations

SA z/OS processor operations monitors and controls target systems with the following operating systems:

- z/OS, OS/390, MVS/ESA, MVS/XA™ (MVS/SP™ V2.2 or higher), z/VM
- VM/SP V6.0, VM/XA V2.1, VM/ESA® V1.1.0
- VSE/SP V4.1, VSE/ESA V1.1.0 or higher
- LINUX of distributions providing Linux for zSeries and S/390 support

**Note:** The above products may no longer be serviced.

## Supported Software

Integrated automation for the following products is supported:

*Table 4. Supported Software*

| | |
|---|---|
| CICS Transaction Server V2.2<br>CICS Transaction Server V2.3<br>CICS Transaction Server V3.1 | For integrated automation of CICS address spaces. |
| IMS Version 7<br>IMS Version 8<br>IMS Version 9 | For integrated automation of IMS address spaces. |
| Tivoli Workload Scheduler for z/OS Version 8.1<br>Tivoli Workload Scheduler for z/OS Version 8.2 | For integrated automation of TWS address spaces. |
| DB2 Version 7<br>DB2 Version 8 | For integrated automation of DB2 address spaces. |

## Customization Dialog Considerations

The SA z/OS customization dialogs do not provide National Language Support (NLS) in the ISPF environment. The SA z/OS customization dialogs must be used with a terminal type of 3278. The terminal type can be set in the Terminal Characteristics portion of the ISPF settings panel.

# Chapter 2. What Is New in SA z/OS 3.2

This chapter contains an overview of the major changes to SA z/OS for Version 3 Release 2. Use this information to check the impact on your user-written programming interfaces, such as automation procedures.

You should also refer to Appendix E, "Migration Information," on page 201 for details of how to migrate to SA z/OS V3.2.

## Enhancements to the Customization Dialog

This section introduces the most important changes in the behavior of the customization dialog for SA z/OS 3.2.

### Nested Class Support

Nested class support allows you to link one application class to another, thus enabling the specification of a class hierarchy. Links from a class can be made in two directions:

- *Downward* to multiple classes or instances, where the policy that is defined in the upper-level class is inherited by the classes or instances
- *Upward* to a single class, so that the lower-level application inherits the policy that is defined in the upper-level class

These classes can carry data on various levels, be nested, and an application can inherit data from a class chain. This can be particularly useful for policy databases with lots of applications of the same type, for example, large IMS installations with lots of IMS subsystems. You might define a top-level class with basic definitions for all IMS applications, and nested classes with more specific definitions for the various IMS subtypes, so that applications can inherit policy values from the top-level class and specific nested classes.

Class policies are inherited down to the instance level where any changes to the policy on that level override the inherited policy. Inherited policy data is initially displayed in a different color but you can change this from the customization dialog Settings Menu.

### Policy Database Import

The policy database import function has been extended to allow you to select GRP and SYS entry types to be imported. You can also import the entry types that are linked to these, the links between resources in the groups or systems, and the resources in the systems.

### Policy Database Flat File Update

The flat file update facility has been enhanced to allow you to create new entries of any type and to update the following:

- The THRESHOLDS policy for the Application (APL) entry type
- All policies for the User E-T Pairs (UET) entry type
- All policies for the Monitor Resource (MTR) entry type
- A subset of policies for the Application Group (APG) entry type

## CHRON Timer Support

This is an extension of the TMR definitions that allows you to define NetView CHRON timer parameters and thus use the full range of command scheduling that is offered by the CHRON command. (For more details, see *Tivoli NetView for z/OS Command Reference Volume 1*.)

Definitions that you make with the TIMERS policy item are:

- Automatically included in the ACF fragment during configuration build
- Added to the CHRON timer definitions when the CHRON timer is created during SA z/OS initialization

## Add-On Sample Policy Databases

Best practice policies (sample policy databases) are delivered with SA z/OS 3.2 that allow you to get productive much faster, also in the case of migrating from competitive products.

The collection of policies that is delivered with SA z/OS has been enhanced, and new policies have been added, such as:

- *HYPERSWAP
- *IBMCOMP
- *ITM (previously *OMEGAMON)
- *TBSM

You can find further information about each of the samples as follows:

- Using the V(iew) command when creating a new policy database or importing from add-on policies
- In the DOCUMENTATION policy item of the ENTERPRISE entry type

Diagrams of the policies are also provided as PDF files that are located in the USS installation path. The default for this path is: /usr/lpp/ing/doc/policies/.

## Sysplex Defaults policy object (XDF)

The Sysplex Defaults policy object allows you to define sysplex resource information defaults.

## Default Desired Status

An option called Desired Available has been introduced that sets the Desired Status of a resource in the absence of any propagated desiredStatus votes. It is available for:

- Subsystems (APLs)
- Application groups (APGs)
- Monitor resources (MTRs)

## Shared WLM Resource Names Support

You can now define a single WLM resource for *multiple* SA z/OS resources. The state of the WLM resource is ON as long as one of these SA z/OS resources is available.

## Extended Status Command Support

Extended status command support allows you to manage components that have not been defined to SA z/OS but still depend on the availability of a subsystem. You can perform actions on applications or application components while certain services are started or stopped, or if they fail. For example, consider TCP/IP communication using WebSphere MQ. The MQ TCPIP Listener component is an application's network interface to the MQ queue manager. It listens to a particular port for TCP/IP connections. When TCP/IP is down, the listener can be stopped because there is then no point listening to the port. However, as soon as TCP/IP is up again, the listener must be restarted and listen to its port again.

Applications, or their components, that consume services are known as *service consumers* (or simply consumers), and applications that provide services are known as *service providers* (or simply providers). Thus the Listener is the service consumer that consumes services that are provided by TCP/IP, which is the service provider.

In certain cases the consumer application knows its provider application at configuration time. This is known as a *static* link between a consumer and a provider. In other cases, the consumer has to evaluate its provider application at run time. This is a *dynamic* link. Note that only consumers and providers on the *same* system can be linked.

## Enhanced Reporting: Inform List

You can define the individual registration of resources through the **Inform List** input field. You can specify:

- Whether a resource (APL, APG, or MTR) is registered with SDF, NMC, the TWS status observer, IOM, or SMF
- Default values for certain inheritance levels for the entry types SDF, XDF, MDF, and ADF (similar to that for the captured messages limit)

The advantages of this include:

- Reducing unnecessary data traffic
- Relaying the right resource information to the right people
- More efficient alerting and notification

If you have a large number of resources this can help improve performance.

**Note:** You have to ensure that a System Defaults Entry with default settings for SDF and NMC exists in order to retain the notification for SDF and NMC, respectively, if you already use any of these. Refer to “NMC Migration” on page 207 for details.

## Availability and Recovery Time Reporting

SA z/OS introduces availability and recovery time reporting. System Management Facility (SMF) records can now be written for subsystems (APLs), application groups (APGs), and monitor resources (MTRs). The data can be used to produce a spreadsheet to assist you in billing users or reporting the reliability of your critical applications or the software that those applications are dependent on.

## Event-Based Monitoring Using Monitor Resources

Event-based monitoring by SA z/OS allows you to explicitly bind monitor resources to real-world objects and optionally jobs. A trigger that contains the monitored object name or the job can then be used by SA z/OS to locate the Monitor Resource and to set the health status or issue commands. This allows SA z/OS to handle a variety of monitoring events. This has been implemented for:

- CICS link and health monitoring
- OMEGAMON XE situations

Message ING150I has been introduced as a trigger that allows you to set the health status of a Monitor Resource based on the monitored object name and job. The monitored object name must be prefixed to be able to identify objects for different products (for example, between CICSPLEX/SM and ITCAM for SOA).

INGMON, the generic routine that is responsible for health monitoring, is invoked from the NetView automation table whenever ING150I is issued. It locates the Monitor Resource for a given monitored object or job and then looks up the code match table for the health status or commands, or both, that should be issued whenever the triggering event occurs.

### CICSPlex Monitoring

Event-based CICS link and health monitoring is implemented using CICSPlex System Manager (CICSPlex SM) objects. Whenever an event is received from CICSPlex SM, message ING150I is issued.

INGCPSM is the event listener for CICSPlex SM. Because it is a long-running CLIST it needs to be run in a virtual operator station task (VOST). It scans the configuration on startup and listens for events. It then periodically checks whether the configuration has changed (that is, Monitor Resources have been added, deleted, or changed, etc.) or Monitor Resources are waiting for initial monitoring (that is, they have STATUS=ACTIVE and HEALTH=UNKNOWN).

### Monitoring OMEGAMON XE Situations

Unlike the exception-based monitoring that SA z/OS uses for classic OMEGAMON monitors, the OMEGAMON XE infrastructure provides the means to react to situations whenever they occur. On the Tivoli Enterprise Portal (TEP), a user can specify what kind of automated response (reflex automation) should be triggered for each individual situation.

SA z/OS makes use of this capability by providing a simple command called INGSIT. This command can entered on the TEP with the Take Action dialog by the ITM administrator for those situations where SA z/OS health monitoring or health-based automation should take place.

The Take Action command is carried out on the agent, in this case, OMEGAMON XE for z/OS, and not the Tivoli Enterprise Monitoring Server (TEMS) unless the TEMS is on the same system. This is because it is possible that the hub TEMS may not reside on z/OS and so the command may not be delivered.

INGSIT triggers message ING150I that allows you to set the health status of individual Monitor Resources. It is then possible to issue commands, such as recovery or notification commands, to automatically fix the situation. You can specify what health status and associated commands are issued in the customization dialog.

## OMEGAMON XE Integration

SA z/OS is now able to communicate with IBM Tivoli Monitoring (ITM) through the standardized Simple Object Access Protocol (SOAP) interface. Although it is not possible to interact with each and every monitoring product individually, as is the case with the existing OMEGAMON classic interface of SA z/OS, monitoring data can be accessed through the hub Tivoli Enterprise Monitoring Server (monitoring server).

The hub monitoring server can be configured such that other hubs may be reached through it. In this case a single connection to one hub monitoring server may offer access to many monitoring products outside the scope of this hub monitoring server.

The SA z/OS utility command INGOMX has been enhanced to read SOAP request data from the default SAFE or from a data set, and connect to a SOAP server that has been defined in the customization dialog, and that returns the SOAP response to the caller that requested it. This SOAP response can be processed easily through standard NetView programming techniques such as PIPE.

## Alert/Escalation via SA IOM

SA z/OS can now access the notification feature of System Automation for Integrated Operations Management (SA IOM) to alert operators or system programmers in case of SA z/OS problems where manual intervention is required.

SA z/OS connects to the SA IOM Server and triggers the notification process for an alert. Automation managers cannot connect to SA IOM so they inform their automation agents whenever application groups that they are responsible for experience problems that require alert notification.

Alerts can be triggered by executing the INGALERT command. This can be either from the NetView automation table, from a CLIST or from the command line.

The customization dialogs allow you to specify IOM for reporting and to define alert points for application groups. SA z/OS provides a set of built-in alert points. You can use them by defining INGALERT as a message ID and specifying appropriate code entries for it. In addition you can use any user-defined alert ID you want. Specify it in the corresponding code entry and call INGALERT with this ID.

Alerting can be turned on or off on at the following levels:

- System, using the INGCNTL command
- Resource, through the Inform List input field
- Alert ID, using code matching

## Enhanced WTOR Processing

The processing of WTORs by SA z/OS has been enhanced to:

- Avoid bottlenecks
- Improve performance
- Make use of NetView PIPEs and DOM messages
- Reduce run time
- Improve serviceability and reliability

When SA z/OS receives WTORs (write-to-operator-with-reply requests), it either automatically replies to them, or stores them if they are to be used for recovery or to shut down the subsystem that issued them. WTORs that are stored for later use are known as outstanding WTORs.

All WTORs that are issued at a system are forwarded to NetView and processed by the NetView automation table (AT) that triggers generic routines according to the processing purpose.

You can use the MESSAGES/USER DATA automation policy item to define what response SA z/OS should make to incoming WTORs for applications, monitor resources and MVS components.

SA z/OS keeps track of all outstanding WTORs that have not yet been replied to and displays them via SDF or NMC. You can use the automation policy to define the severity for outstanding WTORs and a priority that allows you to distinguish between primary and secondary WTORs.

## Automation Flag Processing

Automation flags can be used to control whether or not automation actions are executed by SA z/OS. They can be set at different levels to control specific situations:

- Related to different phases in the lifetime of an application
- Resource or even minor resource specific
- As the default for applications or MVSESA resources
- As the default for a system
- In the automation policy database or during run time
- Permanently or limited in time

Restructured automation flag processing in SA z/OS 3.2 results in:

- Reduced storage consumption
- Reduced usage of timers
- Consistent checking of automation flags for minor resources
- Improved performance

The separate definition of ASSIST flags is no longer possible. To specify the LOG assist mode in SA z/OS 3.2, you can set the automation flag to L, which means that triggered automation actions that this flag is set for, such as commands or replies, are written to the NetView Log instead of being issued.

## Minor Resource Thresholds

SA z/OS integrates the concept of minor resource thresholds into the base code, thus providing a consistent concept for usage by each application. This includes threshold checking with a consistent evaluation concept related to the search sequence for default values of thresholds. Furthermore, SA z/OS 3.2 considers thresholds when issuing actions. This increases the flexibility in defining actions that are dependent on the frequency of the triggering event.

Minor resource thresholds can now be defined using the MINOR RESOURCE THRES policy item for all application types and additionally for MVSESA resources with the MVC entry type. CICS and IMS minor resource thresholds are also now defined with the MINOR RESOURCE THRES policy item. Minor resource names must be specified separately for minor resource flags and minor resource thresholds.

Thresholds can also be specified during run time with the INGTHRES operator command.

When SA z/OS has to control automation actions for events via thresholds, the threshold definitions are retrieved from the automation policy, and the error timestamps for the last events of the same type that have been registered are retrieved from the status file. If the frequency of the event exceeds the critical threshold, only a notification is issued to inform about the situation. If however the critical threshold has not yet been reached, the defined automation action is executed.

## Improved Group Behavior

### Move Mode

You can now define a *move mode* for MOVE application groups so that the order to start the new group is not sent until the old member has become inactive. You can specify whether the new groups is started in:

- **Parallel:** At the same time as the old member is being stopped.
- **Serial:** After the old member has been stopped completely.

### Preemptive Move

Preemptive move allows you to use preference values so that the automation manager can plan an application move when an operator requests a system shutdown. This can lead to much reduced downtime for the system.

## IMS Feature Cleanup

SA z/OS 3.2 integrates the startup, shutdown, recovery, and monitoring of IMS applications into its base functionality. The new structure makes use of existing base functionality and monitor resources (MTRs). This results in the following:

- Less IMS feature code.
- The use of base SA z/OS functions to start and stop resources.
- The use of MTRs to have relationships to the IMS applications. This further makes it possible to have customizable recovery actions from the MESSAGES/USER DATA policy or the HEALTHSTATE policy.

As part of the cleanup, the IMS panels have been modernized to give them the same look and feel as standard SA z/OS command dialogs.

## Work Item Statistics

You can use the INGAMS command dialog to display work item statistics. INGAMS shows history information about the work items processed by the automation manager. The automation manager keeps track of the last 300 work items processed by each of the tasks that build the automation manager kernel.

SA z/OS introduces work item lifecycle recording, an internal diagnostic tool that you should use only if required by SA z/OS service. It provides enhanced debugging to track down lost requests during automation agent-automation manager communication and other automation manager-related problems.

## NMC Enhancements

### Multiple NMC Focal Points

Multiple NMC focal points are supported by focal point specific heartbeats. Typically two NMC focal points are used to survey the NMC targets. Each focal point is the backup of the other one. They always show identical information about the targets (hot backup). Both focal points have to be configured in the SA z/OS customization dialogs. Each focal point needs its own heartbeat (heartbeat timer on target system and deadman timer on focal point system).

### Bulk Status Update

Many resource status changes that occur at the same time are processed together. They are sent to the NMC focal point with one call of INGPOST and applied to the RODM database with one call. This makes processing of these changes faster and more efficient. Thus, for example, the startup of a group with many members is displayed much faster on the NMC client with bulk processing

## Enhancements to SA z/OS Commands

### New Commands

*Table 5. New Commands Delivered with SA z/OS 3.2*

| Command | Where Invoked |
|---|---|
| “DISPAPG” on page 17 | Operator interface |
| “INGMDFY” on page 18 | API |
| “INGCNTL” on page 17 | API |
| “INGQRY” on page 17 | REXX function |
| “INGPSMON” on page 17 | API |
| “INGVMON” on page 17 | API |
| “ISSUEACT” on page 17 | API |
| “INGALERT” on page 17 | API |
| “INGCPSM” on page 18 | API |
| “INGLINK” on page 18 | API |
| “INGMDFY” on page 18 | API |

*Table 5. New Commands Delivered with SA z/OS 3.2 (continued)*

| Command | Where Invoked |
|---|---|
| “INGSIT” on page 18 | API |
| “ING$QRY” on page 18 | AT function |

## DISPAPG

The DISPAPG command displays detailed information about a specified application group.

## INGCNTL

The INGCNTL common routine is used to enable and disable alerting to System Automation for Integrated Operations Management (SA IOM) and to set the parameters that are required for the connection to the SA IOM server.

## INGQRY

The INGQRY common routine returns the value of the specified attribute for a particular resource.

## INGVSTRT

The INGVSTRT common routine allows an automation procedure to start a virtual operator station task (VOST).

## INGVSTOP

The INGVSTOP common routine allows an automation procedure to stop a virtual operator station task (VOST).

## INGPSMON

The INGPSMON monitoring routine is used to determine the status of an MVS subsystem. Unlike INGPJMON it does not search MVS address space control blocks.

## INGVMON

The INGVMON monitoring routine is used to determine the status of a virtual OST (VOST). It should be used as monitoring routine in a VOST management APL.

## ISSUEACT

ISSUEACT, ISSUECMD and ISSUEREP are defined as synonyms for the same generic routine, which can be used to trigger your own commands, replies, or both, from messages that are defined in the automation policy item MESSAGES/USER DATA under consideration of the automation flags.

If the generic routine is called as ISSUECMD, only commands are issued, whereas if it is called as ISSUEREP, only replies are issued. When called as ISSUEACT, the generic routine issues commands and replies according to the given selection criteria that are passed as parameters.

In addition, this generic routine includes special message processing for some critical DB2 messages and for JES2 message $HASP099.

## INGALERT

The INGALERT utility allows you to send alerts to operators or system programmers whenever there is a problem with SA z/OS that requires their attention.

### INGCPSM

The INGCPSM utility returns status information for a CICSPlex® System Manager (CICSPlex SM) object. This data is returned in ING150I messages. INGCPSM should be run in a virtual operator station task (VOST).

### INGLINK

The INGLINK utility lets you:

- Activate and deactivate a link between a consumer and a provider application that is defined as a dynamic link in the automation policy
- Query the status of a link between a consumer and a provider application as defined in the automation policy

A consumer is an application that has messages defined for it with a prefix of UP_ or DN_ and with a suffix of a valid subsystem name.

A provider is an application whose subsystem name is used as the suffix for an UP_ or DN_ message that is defined for any application.

### INGMDFY

The INGMDFY utility displays the defined actions for the startup or shutdown of a subsystem that are currently loaded and allows you to modify them for the next startup or shutdown. INGMDFY also allows you to define additional actions or to delete defined actions for the startup or shutdown of a subsystem.

### INGSIT

The INGSIT utility allows you to report an event to SA z/OS. To allow SA z/OS to react to that event, it must be associated with one or more Monitor Resources through the monitored object name and optionally a job name. The event information consists of the monitored object name, an optional event severity, an optional job name, and optional related data. INGSIT transforms the event into a well-formed ING150I message that can subsequently be automated on behalf of the monitored object according to the definitions in the automation policy.

### ING$QRY

SA z/OS provides a NetView Automation Table Function (ATF), called ING$QRY. This allows you to query or compare the status and other important attributes of jobs that are controlled by SA z/OS from within the AT and use the result as a condition in the AT statement. ING$QRY is an alias. The routine returns the attribute or comparison result as the function value of the NetView ATF function so that it can be used within the AT statement

## Enhanced Commands

**DISPINFO**

The DISPINFO command has been modified to show:

- The inform list
- The subcategory of the resource
- The class chain list, if applicable

**DISPMTR**

The DISPMTR command has been modified to show

- The timestamp, in chronological order, of the last startup and shutdown of the monitor
- The Inform list for the monitor

**INGAMS**

You can use the SUSPEND and RESUME parameters to control the sending of orders to an automation agent by the automation manager (in case, for example, an order to shut down a system has been unintentionally sent).

**INGAUTO**

You can have INGAUTO write commands and replies to the netlog if an event occurs that triggers an automated action.

**INGGROUP**

A new action called OVERRIDES has been introduced for the INGGROUP command. If specified, it displays resource groups whose attributes have been modified by means of the INGGROUP command.

**INGINFO**

The information displayed by INGINFO has been changed as follows:

- The **Compound Status** now has a timestamp
- The **Desired Available** setting is shown, which is the Desired Status of the resource in the absence of any propagated desiredStatus votes

**INGMOVE**

INGMOVE now supports the moving of a sysplex application group to another system.

**INGPLEX SVCdump**

The INGPLEX SVCdump command can now be executed in line mode.

**INGREQ**

You can now specify a feedback parameter that causes the final result of the command to be reported back to a designated instance.

The INGREQ panel has been redesigned so that **Comment** field is on the first screen.

**INGSTOBS**

A new wait option parameter has been added to the INGSTOBS command. It controls the maximum time to wait for process completion when a request has been sent to the automation manager, for example, when subscribing a user-specified exit as the status change observer.

**INGTIMER**

You can now specify a day of the week when coding an interval with the EVERY parameter.

**RESYNC**

The RESYNC command now lists explicitly the resources that it can be used for.

At the end of resynchronization, subsystems in a DOWN or RESTART status are not automatically started or restarted.

**SETTIMER**

The SETTIMER command has been enhanced so that you can define either an AFTER, AT or EVERY timer, and also additional options that apply to a CHRON timer.

## Enhancements to Processor Operations

### Password Support for ISQCCMD ,CBU Command

With CBU password support, ProcOps or BCPii users can directly enter the password together with the CBU 'ACTIVATE' or 'TESTACT' command.

### BCPii Support for ProcOps Lpar Management Commands

The SA z/OS BCP Internal Interface (BCPii) hardware interface now offers a common set of hardware commands (using ISQCCMD) to manage and control the logical partitions of your System z™ and zSeries processor hardware. In addition, the management of processor activation profiles and queries of CPC and LPAR information is available.

## Enhancements to I/O Operations

### I/O Operations Supports TCP/IP Communication

With SA z/OS V3.2 I/O operations supports TCP/IP in parallel to VTAM. All I/O operations applications running SA z/OS V3.2 will try to communicate using TCP/IP. If this fails for whatever reason the communication protocol falls back to VTAM. However, to allow I/O operations to communicate via TCP/IP you need to define at least the server port (see “Step 18C: Perform TCP/IP Definitions” on page 128).

I/O operations supports a PARM parameter on startup that allows you to restrict the communication to a specific protocol or a particular TCP/IP address space (see “Step 33: Customizing I/O Operations” on page 155).

### I/O Operations Makes Use of the MVS Component Trace

With SA z/OS V3.2 I/O operations has replaced its internal trace table and the GTF trace with the MVS Component Trace. The benefits of this are:

- More information can be traced by a single trace entry
- The trace data can be visualized much more easily
- All I/O operations programs can perform traces even if they are running outside the I/O operations address space as IHVAPI

The implementation requires the I/O operations address space to be non-swappable (see “Step 4B: Update SCHED*xx*” on page 82 and “Step 11: Customizing the Component Trace” on page 111).

### Renumbering of I/O Operations Messages

New display messages have been introduced due to the new communication via TCP/IP. The current range of message numbers in this area has been exhausted and has required the reservation of a new range from IHVC200 through IHVC299 for these messages.

Because of this, message numbers that represent hexadecimal values have been retired, especially in this area, and converted to decimal values, as shown in Table 6 on page 21.

*Table 6. Renumbered I/O Operations Messages*

| Old | New | Old | New | Old | New |
|---|---|---|---|---|---|
| C53A | C540 | O00A | O010 | V13A | V151 |
| C80A | C200 | O00B | O011 | V13B | V152 |
| C80B | C201 | O00C | O012 | V13C | V153 |
| C80C | C202 | O00D | O013 | V13D | V154 |
| C80E | C204 | O00E | O014 | V13E | V155 |
| C80F | C205 | O00F | O015 | V13F | V156 |
| C81A | C210 | O010 | O016 | V23E | V236 |
| C81B | C211 | O011 | O017 | | |
| C81C | C212 | O012 | O018 | | |
| C81D | C213 | O013 | O019 | | |
| C81E | C214 | O014 | O020 | | |
| C81F | C215 | O015 | O021 | | |
| C82A | C220 | O016 | O022 | | |
| C82B | C221 | O017 | O023 | | |
| C82C | C222 | O018 | O024 | | |
| C82D | C223 | O019 | O025 | | |
| C82E | C224 | O020 | O032 | | |
| C82F | C225 | O021 | O033 | | |
| C83A | C230 | O030 | O048 | | |
| C83B | C231 | O040 | O064 | | |
| C83C | C232 | O041 | O065 | | |
| C83D | C233 | O0FC | O252 | | |
| C83E | C234 | O0FD | O253 | | |
| C83F | C235 | O0FF | O255 | | |
| C84A | C240 | | | | |
| C84B | C241 | | | | |
| C84C | C242 | | | | |
| C84D | C243 | | | | |
| C84E | C244 | | | | |
| C87A | C270 | | | | |
| C87B | C271 | | | | |
| C87C | C272 | | | | |

# Chapter 3. Planning to Install SA z/OS on Host Systems

## Component Description

The SA z/OS product consists of the following components:

- System operations (*SysOps* for short)
- Processor operations (*ProcOps* for short)
- I/O operations (*I/O Ops* for short)

### System Operations

System operations monitors and controls system operations applications and subsystems such as NetView, SDSF, JES, RMF™, TSO, RODM, ACF/VTAM®, TCP/IP, CICS, DB2, IMS, TWS, OMEGAMON® and WebSphere.

Enterprise monitoring is used by SA z/OS to update the NetView Management Console (NMC) resource status information which is stored in the Resource Object Data Manager (RODM).

### Processor Operations

Processor operations monitors and controls processor hardware and VM guest systems operations. It provides a connection from a focal point processor to a target processor. With NetView on the focal point processor, processor operations automates operator and system consoles for monitoring and recovering target processors.

Processor operations allows you to power on and off multiple target processors and reset them. You can perform IPLs, set the time of day clocks, respond to messages, monitor status, and detect and resolve wait states.

### I/O Operations

I/O operations provides a single point of control for managing connectivity in your active I/O configurations. It takes an active role in detecting unusual I/O conditions and lets you view and change paths between a processor and an input/output device, which can involve using dynamic switching (the ESCON switch).

I/O operations changes paths by letting you control channels, ports, switches, control units, and input/output devices. You can do this via ISPF dialogs, as well as on an operator console or API.

## SA z/OS and Sysplex Hardware

When SA z/OS is used in a Parallel Sysplex® environment, the hardware setup can be similar to the one illustrated in Figure 1.

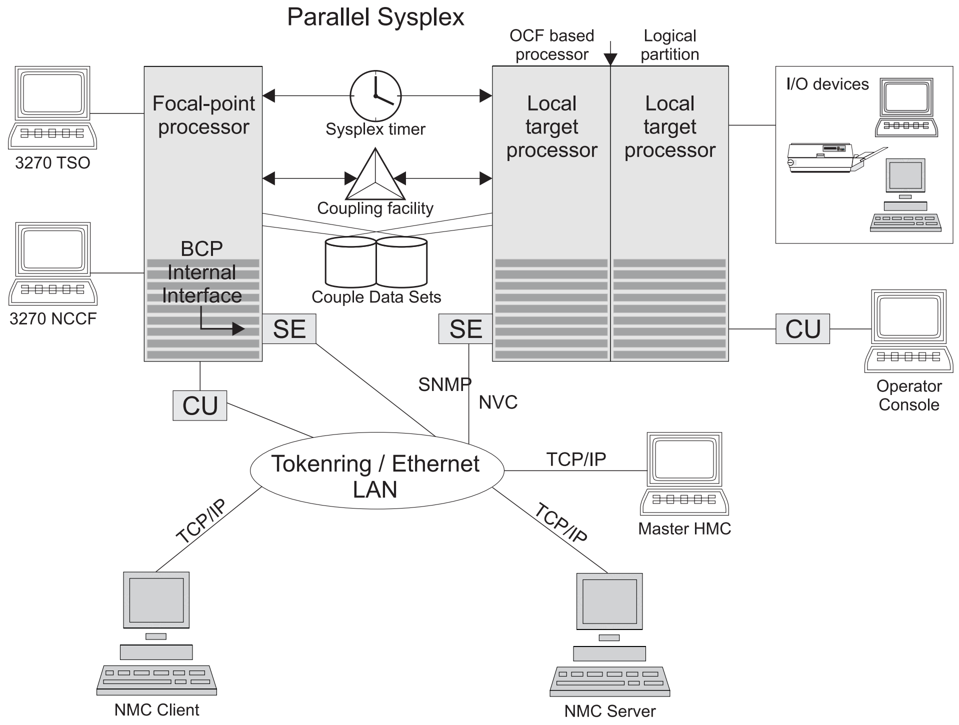


*Figure 1. Basic Hardware Configuration*

It shows a two processor Parallel Sysplex configuration, with systems running on it. One is playing the role of a SA z/OS focal point. For example, the role of the SA z/OS NMC focal point with information about all the systems and applications in the sysplex, running under the control of SA z/OS.

Operators can use a workstation with the SA z/OS NMC client code installed, to work with graphical views of the SA z/OS controlled resources stored on the focal point. The NMC server component receives status changes from the NMC focal point and distributes them to the registered clients to update their dynamic resource views. Sysplex specific facilities, like the coupling facility hardware can be managed and controlled using the NMC's client graphical interface, as well as the 3270 NCCF based SA z/OS operator interfaces.

Operators can also use SA z/OS Tivoli Enterprise Portal (TEP) support to monitor the status of automation on z/OS systems and z/OS sysplexes from a workstation that has a TEP client installed on it.

With the same interfaces, processor operations, another SA z/OS focal point function can be operated. With processor operations it is possible to manage and control the complete processor hardware in a sysplex. Operator tasks like re-IPLing a sysplex member, or activating a changed processor configuration can be accomplished. Processor operations uses the processor hardware infrastructure, consisting of the CPC Support Element (SE), or the Hardware Management Console (HMC) interconnected in a processor hardware LAN, to communicate with the own, other local, or remote located Support Elements of other CPCs. The Support Elements provide the Systems Management Interface OCF (Operations Command Facility) to perform hardware commands like LOAD or SYSTEM RESET to control the hardware and hardware images. SA z/OS processor operations can be customized to use SNA-based NetView connections (NVC), or IP based SNMP for communication. For Parallel Sysplex environments, SA z/OS provides an additional processor hardware interface, the BCP (basic control program) internal interface. This interface is independent from processor operations. It allows processor hardware operation in a sysplex, without requiring external network CUs (control units). From a system in the sysplex, the SE of the own CPC as well as the SEs of the other processors in the sysplex can be accessed.

The following describes some relevant resources used by SA z/OS and its components.

## OCF-Based Processor

A central processor complex that interacts with human operators using the interfaces provided by the Support Element (SE). OCF-based processors are processors from the 390-CMOS processor family.

## Parallel Sysplex

A set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components (coupling devices and sysplex timers) and software services (couple data sets). In a Parallel Sysplex, z/OS provides the coupling services that handle the messages, data, and status for the parts of a multisystem application that has its workload spread across two or more of the connected processors. Sysplex timers, coupling facilities, and couple data sets containing policy and states for basic functions are all part of a Parallel Sysplex. You can control a Parallel Sysplex by NetView-based commands or through an NMC workstation.

## Coupling Facility

A hardware storage element with a high-speed cache, list processor, and locking functions that provides high performance random access to data for one system image or data that is shared among system images in a sysplex. With I/O operations you can see standalone coupling facilities. It handles them as control units with up to eight devices, all defined by the user. With SA z/OS system operations, you can display the status of coupling facilities from a single system's point of view or you can display sysplexwide status.

## Sysplex Timer

An IBM unit that synchronizes the time-of-day (TOD) clocks in a multiprocessor or in processor sides. External Time Reference (ETR) is the generic name for the IBM Sysplex Timer® (9037).

## Logically Partitioned (LPAR) Mode

A processor with the Processor Resource/Systems Manager™ (PR/SM™) feature that can be divided into partitions with separate logical system consoles that allocates hardware resources among several logical partitions. (It is called *logical* because the processor is not physically divided, but divided only by definition.) The partitions are defined, monitored, and activated separately by processor operations.

A processor that does not use logical partitions is in “basic mode”.

## Communications Links

Links that connect the focal point processor to target processors so that commands, messages, and alerts can flow. For more information refer to “Defining System Operations Connectivity” on page 53.

### NetView Connection (NVC)

SNA-based communication between the processor operations focal point and the operator control facility (OCF), which runs on the Support Element (SE). For this connection, processor operations uses the NetView RUNCMD interface and the NetView FOCALPT command.

### SNMP

Alternative to NetView connections, SNMP may be chosen as the protocol for communications between the processor operations focal point and the OCF of an SE.

See also “Understanding the Processor Operations SNMP Interface” on page 28.

### BCP Internal Interface

For processor hardware automation in a sysplex environment, this link allows an OS/390 or z/OS system directly to communicate with the OCF of its own hardware SE, as well as the OCFs of other hardware SEs which are part of a cluster of processors. This cluster must be defined to the Master HMC in a processor environment. If a sysplex processor hardware is to be automated, the processor hardware of all sysplex members must be defined to the Master HMC.

See also “Understanding the BCP Internal Interface” on page 27.

### NetView RMTCMD Function

A connection that allows communication between the target and focal point system in order to pass status changes to the focal point system. This communication method is also used for other purposes.

### TCP/IP

For VM second level system automation, this link allows SA z/OS ProcOps to communicate with the ProcOps Service Machine (PSM) on the VM host of the second level systems.

See also “Understanding the TCP/IP Interface” on page 29.

## Control Units (CU)

Control units are hardware units that control input/output operations for one or more devices. You can view information about control units through I/O operations, and can start or stop data going to them by blocking and unblocking ports. For example, if a control unit needs service, you can temporarily block all I/O paths going to it.

## I/O Devices

Input/output devices include hardware such as printers, tape drives, direct access storage devices (DASD), displays, or communications controllers. You can access them through multiple processors. You can see information about all devices and control paths to devices. You can vary devices or groups of devices online or offline.

## NetView Management Console (NMC)

A NetView function that consists of a graphic series of windows controlled by the NetView program and that allows you to monitor the SA z/OS enterprise interactively. The NetView Management Console consists of an NMC server and an NMC client.

The NMC client is connected to the NMC server that communicates with NetView. The NetView Management Console (NMC) can be implemented with an optional client, either on the server or separately.

## Tivoli Enterprise Portal Support

SA z/OS Tivoli Enterprise Portal (TEP) support allows you to monitor the status of automation on z/OS systems and z/OS sysplexes using a TEP client. The client is the user interface for an SA z/OS monitoring agent. The monitoring agent uses Tivoli Monitoring Services infrastructure, which provides security, data transfer and storage, notification mechanisms, user interface presentation, and communication services for products in the IBM Tivoli Monitoring and OMEGAMON XE suites in an agent-server-client architecture.

The monitoring agent is installed on the systems or subsystems in the sysplex that you want to monitor and passes data to a hub Tivoli Enterprise Monitoring Server (monitoring server), which can be installed on z/OS, Windows, and some UNIX operating systems. The monitoring server communicates with the Tivoli Enterprise Portal Server (portal server), which then communicates with the portal client.

For more details, see *IBM Tivoli System Automation for z/OS Monitoring Agent Configuration and User's Guide*.

# Planning the Hardware Interfaces

This section provides additional information about the processor hardware interfaces supported by SA z/OS.

## Understanding the BCP Internal Interface

In order to allow the sysplexwide activation or deactivation of the coupling facilities and to control sysplex members leaving the sysplex, SA z/OS uses the BCP (Basic Control Program) internal interface. The BCP internal interface of the following processor hardware families is supported:

- Series z9

- zSeries
- CMOS-S/390 G6
- CMOS-S/390 G5

Using the BCP internal interface from MVS allows you to send hardware operationscommands such as SYSTEM RESET, or ACTIVATE to the Support Element attached to its own processor hardware (CPC). If the CPC is configured in LPAR mode, the operations command can be sent to all logical partitions defined on the CPC.

Furthermore, with the enhanced sysplex functions of SA z/OS, sysplex members running on other CPCs than their own image can be controlled through the BCP internal interface. This is possible by defining all CPCs of your sysplex on the master HMC of your processor hardware LAN.

The following processor hardware can be controlled as a target with the BCP internal interface from the above listed processors, but cannot use the SA z/OS BCP internal interface to control itself or other processors:

- CMOS-S/390 G4
- CMOS-S/390 G3

At the processor hardware LAN level, the BCP internal interface uses the SNMP transport protocol. For this reason, the Support Elements need to be customized for SNMP. One HMC in the processor LAN must be configured to be the Change Management Master HMC, otherwise routing between the own SE and other SEs will not work.

Note that the MVS/HCD function uses the BCP internal interface to update IOCDS and IPL information in the Support Elements of addressed CPCs. You cannot use SA z/OS to perform these tasks, nor can HCD be used to perform the hardware operations functions of SA z/OS.

Currently, the BCP internal interface cannot be used by the processor operations focal point application. The interface can be configured and used for Parallel Sysplex automation purposes only. Exceptions to this are are the processor operations common commands for LPAR management, see “Chapter 5. Common Commands” in *IBM Tivoli System Automation for z/OS Operator’s Commands* for details.

## Understanding the Processor Operations SNMP Interface

Using the SNMP interface of processor operations, you can monitor and control local or remote processor hardware from a processor operations focal point NetView in an IP network environment. This is different to the BCP internal interface, which allows mutual hardware control among sysplex members without a system network dependency.

With the processor operations SNMP interface, the following processors can be managed:

- Series z9
- zSeries
- CMOS-S/390 G1 through G6
- Multiprise® 3000

As with the BCP internal interface, its purpose is to support the OCF commands (for example, ACTIVATE, SYSRESET) provided by the processor hardware.

The Support Elements of the CPCs you want to control must be configured for SNMP. Alternatively, you can configure a single HMC instead of multiple Support Elements in your processor LAN environment for SNMP. On this HMC the CPCs you want to control must be defined. Multiple HMCs, SEs, or both can be defined in your SA z/OS configuration.

Because this interface uses the IP network for communication between the processor operations focal point and the SEs or HMCs, the TCP/IP UNIX System Services stack is required to be active on the processor operations focal point system.

## Understanding the NetView Connection (NVC) of Processor Operations

Using the NVC interface of processor operations, you can monitor and control local or remote processor hardware from a processor operations focal point NetView in an SNA network environment. With a NVC, the Support Elements must be configured with a valid CPC SNA address. At least one HMC in your processor hardware LAN, where the addressed CPCs are defined, must have a Problem- and Operations Management SNA gateway defined.

As with the other interfaces, a NVC connection can be used to perform OCF requests supported by the processor hardware. The following processor hardware can be configured for NVC:

- zSeries
- CMOS-S/390 G1 through G6
- Multiprise 3000
- Multiprise 2000
- Application Starter Pak

## Understanding the TCP/IP Interface

Using the TCP/IP interface of Processor Operations, you can monitor and control VM guest systems from a Processor Operations focal point NetView in an IP network environment.

Processor Operations communicates with the ProcOps Service machine (PSM) using TCP/IP. The PSM can be regarded as an HMC or SE substitute for the virtual machines. The PSM itself uses the VM/CP Secondary Console InterFace (SCIF) facility to communicate with the single VM second level systems.

The TCP/IP UNIX System Services stack is required to be active on the Processor Operations focal point system.

## Deciding Which Hardware Interface to Use

If you want to use the Parallel Sysplex enhancements of SA z/OS and you have configured your customization to use IXC102A message automation, the BCP internal interface is required.

Note, that this interface can coexist with the supported SNMP and NVC interfaces on a processor operations focal point system. Because the IXC102A automation, which is part of the Parallel Sysplex XCF automation, can also be performed in

SA z/OS using proxy resources together with processor operations, a decision must be made, which automation to use. It is recommended to use the XCF automation based on the BCP internal interface and to disable the IXC102A proxy resource automation based on processor operations.

The following criteria are important for planning which processor hardware interface you can use with processor operations:

- Processor hardware LAN
- Processor hardware type

Only if your processor hardware LAN is token-ring you can use NVC. SNA based NetView connections with an Ethernet LAN are not supported by the Support Elements. However, a token-ring based processor LAN can be used for both NVC and SNMP connections. If your processor hardware LAN has an Ethernet LAN, SNMP must be used.

From the list of the supported processor hardware, only the zSeries models z900 and z800 support SNA based NetView connections. Later zSeries hardware models will support SNMP connections only.

# NetView Data Sets Used by SA z/OS

This section provides information on:

- "SA z/OS Partitioned Data Sets"
- "Shared Data Sets" on page 31

## SA z/OS Partitioned Data Sets

You may need data sets for the DSIPARM, DSIMSG, DSICLD, DSILIST, CNMPNL1, STEPLIB, and DSIPRF concatenations. These data sets are concatenated in various data definition (DD) names in the NetView startup procedure.

There are two types of data set that you may need to customize:

- "NetView Data Sets"
- "SA z/OS-Specific Data Sets" on page 31

### NetView Data Sets

**SA z/OS DSILIST**

This data set holds the Automation Table (AT) load protocol. When ATs that were generated by the SA z/OS build process are loaded, a load protocol is stored in the DSILIST data set. This protocol is required to inspect the AT loaded as INGMSG02.

This can be controlled by the AOFMATLISTING advanced automation option (AAO).

For details about the AT build process, refer to *IBM Tivoli System Automation for z/OS Customizing and Programming* and *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

**SA z/OS DSIPARM**

This data set holds SA z/OS NetView definitions that are specific to this NetView. You may need it only for the DSIPARM concatenation. It should contain the NetView style sheet specifications that define this domain and a suitably customized AOFMSGSY member.

. Depending on your NMC setup, it should contain your NMC specifications, such as the INGTOPOF file and the corresponding BLDVIEWS members. For information about this customization, see *IBM Tivoli System Automation for z/OS Customizing and Programming*.

### SA z/OS-Specific Data Sets

**SA z/OS ACF**

This data set holds the various automation control file fragments and control file fragments for the automation manager needed by the SA z/OS instance on this NetView. It is not recommended that you build your SA z/OS policy databases directly into this data set. Changing your active SA z/OS policy is a process that should be under change control.

Consequently it is recommended that you allocate a separate partitioned data set for your policy database, your ACF fragments created from the ISPF dialogs BUILDF process and the enterprise wide ACF fragments data set specified to the automation manager.

**Notes:**

1. If you use MEMSTORE to load the NetView PDS members in storage (this is the default in NetView) and you do not reload the members after an ACF build, you will get the following message due to an INGAMS command to refresh the configuration:

```
AOF618I ... ACF Token mismatch ...
```

**SA z/OS CREXX**

Compilation of SA z/OS REXX code is optional and supported. If you choose to compile the CLISTs in the SA z/OS SMP/E target data set, this is where the compiled versions should be placed. It is needed only in the DSICLD concatenation for system operations and processor operations REXX CLISTs. If you have compiled I/O operations execs, the data set with these compiled execs must be in the SYSEXEC concatenation.

**Note:** If you use this data set, you can omit the SA z/OS SMP/E target data set in your DSICLD concatenation, because all the code is in it as well.

## Shared Data Sets

By using shared DASD, you are able to reduce the DASD required to store the data sets, but this exposes you to additional risk. With shared DASD you have only one copy of the data sets. As a result, if that DASD volume becomes unusable, you lose access to the data set on ALL the systems that were sharing it. In a large sysplex this may represent a significant operational exposure.

One solution is to have a set of standby procedures for SA z/OS. These are copies of your normal SA z/OS procedures that point to a copy of your data set on another DASD volume and preferably on a different string of DASD volumes. Although an instance of SA z/OS started from these procedures would not share status information with the SA z/OS from your primary procedures, the standby procedures let you maintain operability of your systems in the event that your primary procedures are unavailable.

# REXX Considerations

## Allocation Requirements for REXX Environments

Before running SA z/OS you may need to change the maximum number of REXX environments allowable.

The number of REXX environments allowable is defined in the REXX environment table. See *z/OS TSO/E Customization* for more information. TSO/E provides a SYS1.SAMPLIB member called IRXTSMPE, which is an SMP/E user modification to change the maximum number of language processor environments in an address space. Define the number of allowable REXX environments on the IRXANCHR macro invocation:

```
IRXANCHR ENTRYNUM=xxx
```

For more details, see "Step 14: Verify the Number of available REXX Environments" on page 119

Install the user modification by following the instructions in *z/OS TSO/E Customization*.

# z/OS Considerations

## Prefixes

You should make sure you do not have any load modules, REXX parts or members with the following prefixes:

- AOF
- EVE
- EVI
- EVJ
- HSA
- IHV
- ING
- ISQ

## Defining the XCF Group

In order to be able to communicate in certain situations, the automation manager instances and the automation agents belonging to one sysplex must be members of one and the same XCF group. The name of this group consists of a fixed main part and a variable suffix; the format is INGXSG*xx*. The suffix must be specified separately for the manager and the agents. For the automation manager, it is specified in the HSAPRM*xx* member of SYS1.PARMLIB (see "Step 10A: Customizing HSAPRM*xx*" on page 109); for the automation agents, it is defined in the INGXINIT member of DSIPARM (see "INGXINIT" on page 92).

Systems with SA z/OS NetView instances that belong to the same XCF group must be defined in the Customization Dialogs in the same Group Policy Object of type sysplex. For details refer to the "Group Policy Object" chapter in *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

Note that SA z/OS NetView instances that belong to the same XCF group must reside on different systems. Thus, when you run an SA z/OS 3.1 and an SA z/OS 3.2 agent on the same system, they must not belong to the same XCF group.

### Using SA z/OS Subplexes

You can divide your real sysplexes into several logical SA z/OS *subplexes* (an example is shown in Figure 2). To do this you must define a specific XCF group suffix and a specific group policy object for each subplex. Each SA z/OS subplex must have its own automation manager. In each subplex there must also be only one shared automation manager takeover file and one shared schedule override file.

With SA z/OS subplexes you can run automation on systems of sysplexes in the same way as on single systems. This is required if you do not have shared DASDs for all your systems in the sysplex.

The group ID must be defined in an HSA parmlib member or INGXINIT for NetView.

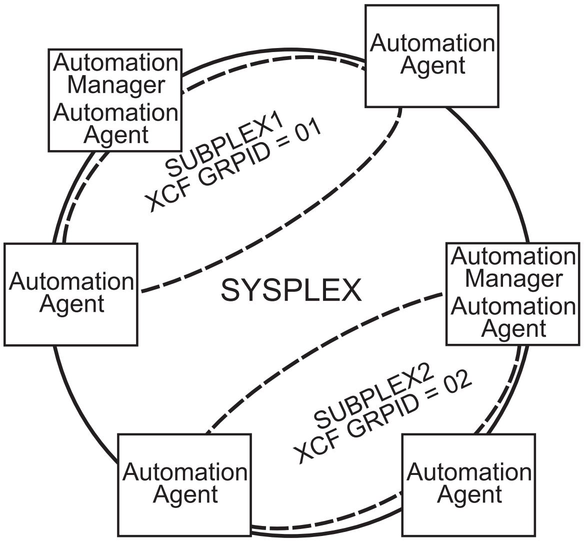


*Figure 2. Using SA z/OS Subplexes*

## System Operations Considerations

NetView ships two sample automation operators, AUTO1 and AUTO2. SA z/OS assumes that these tasks are available and have not been renamed. If they have been renamed, you must change the names in AOFMSGSY and the NetView style sheet, residing in the DSIPARM data set.

## Processor Operations Considerations

It is recommended that you have system operations and processor operations installed together on the same system.

# Automation Manager Considerations

This section presents automation manager considerations relevant to the installation process. For automation manager concepts that are of interest from an operator's point of view, refer to *IBM Tivoli System Automation for z/OS User's Guide*.

The automation manager is introduced as a separate address space. An installation requires one primary automation manager and may have one or more backups. The automation manager is loaded with a model of the sysplex when it initializes. It then communicates with the automation agents in each system, receiving updates to the status of the resources in its model, and sending orders out to the agents as various conditions in the model become satisfied.

A series of substeps is required to get the automation manager up and running for your SA z/OS installation. These installation steps are described in this documentation, but are not identified as being specific automation manager installation steps.

Only the default installation of UNIX System Services is a prerequisite for the automation manager. No hierarchical file system (HFS) or UNIX shell is required.

The automation manager must be defined by RACF (or an equivalent security product) as a *super user* for UNIX System Services. The user that represents the started tasks in your installation must be authorized for the OMVS segment.

**Note:** The system on which the automation manager should be started must be defined as policy object System in the policy database that will be used to create the automation manager configuration file that this automation manager uses (see also "Step 17A: Build the Control Files" on page 123).

## Storage Requirements

When the automation manager is started, it needs a constant amount of storage of 56 MB plus a variable part that depends upon the number of resources to be automated.

The constant part consists of 40 MB for the automation manager code and 16 MB for history information. The rule of thumb for the variable part is `n * 8 KB` where ***n*** is the number of resources.

The sum of storage requirement according to the rule of thumb is:

```
40 MB + 16 MB + n * 8 KB
```

This formula covers the maximum storage requirements. However, the storage requirements does not increase linearly with the number of automated resources. Real measurements may be smaller than values retrieved with the rule of thumb formula.

## OMVS Setup

Because the automation manager requires OMVS, OMVS must be customized to run without JES. (This means that OMVS should not try to initialize colony address spaces under the JES subsystem as long as JES is not available.) Therefore the definitions in the BPXPRM*xx* member must match *one* of the following:

- Either all FILESYSTYPE specifications with an ASNAME parameter are moved into a separate BPXPRM member. This can be activated via the automation policy by using the SETOMVS command after the message `BPXI004I OMVS INITIALIZATION COMPLETE` has been received.
- Or the parameter `'SUB=MSTR'` is added to the ASNAME definition, for example:

```
/*******************************************************/
/* ZFS   FILESYSTEM                                    */
/*******************************************************/
  FILESYSTYPE TYPE(ZFS) ENTRYPOINT(IOEFSCM)
        ASNAME(ZFS,'SUB=MSTR')
```

**Note:** In order to initialize without JES, the Automation Manager needs to be defined as a superuser. If you use an OEM security product that does not initialize until JES has initialized, superuser authority cannot be evaluated until JES is up and consequently JES cannot be started by SA z/OS.

## Recovery Concept for the Automation Manager

For sysplexwide and single-system automation, the continuous availability of the automation manager is of paramount importance.

To ensure the automation manager's functionality as automation decision server, the primary automation manager (PAM), must be backed up by additional automation manager address spaces called secondary automation managers (SAMs). Secondary automation managers are able to take over the function whenever a primary automation manager fails.

Therefore, it is recommended that you have at least one secondary automation manager running. For sysplexwide automation, the SAM should run on a different system than the PAM.

To enable software or hardware maintenance in the sysplex, SA z/OS supports a command to force the takeover of the primary automation manager.

A takeover is only possible when the following requirements are met:

- All the automation manager instances must have access to a shared external medium (DASD) where the following is stored:
  - The configuration data (result of the ACF and AMC build process).
  - The schedule overrides VSAM file.
  - The configuration information data set — this is a mini file in which the automation manager stores the parameters with which to initialize the next time that it is started WARM or HOT.
  - The takeover file.
- If WebSphere MQ is used for communication between the automation manager and the automation agents (see "Manager-Agent Communication and Status Backup" on page 36), all of the automation manager instances must have access to the coupling facility that contains the automation status queue.
- If WebSphere MQ is used for communication between the automation manager and the automation agents, the automation agents must have access to the coupling facility that contains the agent and the work item queues.

SA z/OS follows the concept of a floating backup because:

- The currently active automation manager has no awareness of the existence (and location) of possible backup instances.

- The location of the backup instances can change during normal processing without any interruption for the active automation manager.
- There is no communication between the primary automation manager and its backup instances during normal operation except when a SAM that is to become the new PAM informs the current PAM of that fact during a planned takeover.

This has the advantage that in normal operation, the processing is not impacted by a backup structure which can change.

Depending on the number of resources, the takeover time from a primary to a secondary automation manager is in the range of one to two minutes.

## Manager-Agent Communication and Status Backup

SA z/OS provides two options for establishing communication between the automation manager and the automation agents, and keeping a backup copy of the status of the automated resources, using:

- XCF for communication and a VSAM data set (the takeover file) for backup
- WebSphere MQ V5.3 queues for both communication and backup

For the WebSphere MQ solution, you need WebSphere MQ V5.3 or higher, and also DB2 in data sharing mode. WebSphere MQ V5.3 is required because the PAM and the SAMs must share the queues; this data sharing capability, which is implemented by means of coupling facility list structures, is shipped with WebSphere MQ V5.3. DB2 is required because it serves as the repository for the definitions of the shared queues.

**Note:** SA z/OS also supports a local system environment with WebSphere MQ. In that case, DB2 is not required, but the scope of automation is limited to a single system.

### Using XCF only

XCF communication and the takeover file are primarily intended to substitute for the WebSphere MQ queues in certain situations. However, you can also use them permanently and thus dispense with WebSphere MQ altogether. If you want to do so, you must set the COMM parameter in the HSAPRM*xx* member accordingly; see Appendix G, “Syntax for HSAPRM00,” on page 247.

As already pointed out, the work items and orders to the automation agents that are pending at takeover time are not stored in this implementation, so all these pending items will be lost when the PAM fails and a SAM takes over.

Figure 3 on page 37 illustrates the timeline from the start of the automation manager (AM) through to its termination for the following cases:

- A planned stop and start of the automation manager
- An unexpected failure

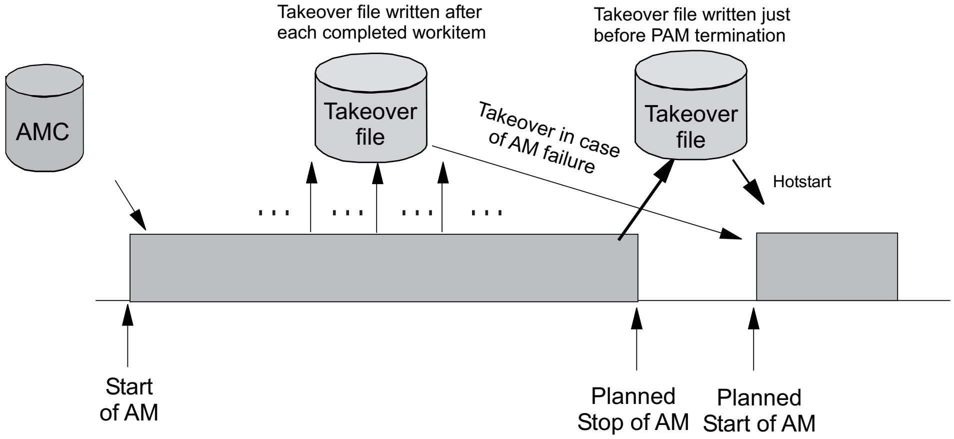

Takeover file written after each completed workitem
Takeover file written just before PAM termination
AMC
Takeover file
Takeover in case of AM failure
Takeover file
Hotstart
Start of AM
Planned Stop of AM
Planned Start of AM

*Figure 3. Using Only the Takeover File for Status Backup*

Table 7 outlines the various recovery scenarios.

*Table 7. Recovery Scenarios*

| Event | SA z/OS Recovery Action | Comments |
|---|---|---|
| PAM fails | SAM runs a takeover | The takeover file contains the state with the last successfully processed work item |
| PAM detects a severe error condition | PAM terminates and SAM runs a takeover | The takeover file is used to rebuild the resource object structures in case of a takeover or next hotstart |
| System with the PAM fails | SAM runs a takeover | The takeover file is used to rebuild the resource object structures in case of a takeover or next hotstart |

## Setting up WebSphere MQ V5.3 (Optional)

If you choose the WebSphere MQ option, the automation manager communicates with the automation agents through two WebSphere MQ queues, and uses a third WebSphere MQ queue for status backup:

- **Work Item Queue:** This queue is the inbound queue for the automation manager. The automation agents put their requests or queries in the form of a work item into this queue. Its name is WORKITEM.QUEUE.
- **Agent Queue:** This queue is the outbound queue of the automation manager. All orders for the automation agents that result from a request (a work item) sent to the automation manager are placed in this queue by the automation manager. The automation agents then pick up the orders from this queue for execution. Its name is AGENT.QUEUE.
- **Automation State Queue:** This queue is only used by the automation manager. It is used to save the current state as well as other information about the resources managed by the automation manager. It is the *automation state queue* that allows SA z/OS to perform a hot takeover, because this queue always contains a consistent image of the resource data that the PAM maintains in storage. Any updates that are made to the resources are also reflected in the automation state queue. Its name is STATE.QUEUE.

The transactional behavior of WebSphere MQ ensures that these three queues are always consistent. A change in any queue is only committed after the corresponding changes have also been made in the other two queues. Thus, for example, the deletion of a work item from the Work Item Queue is only committed when:

- The resulting orders to the agents have been written to the Agent Queue, and
- The resulting state changes of the affected resources have been written to the Automation State Queue.

Figure 4 shows how the queues interact with the automation manager and the automation agents.

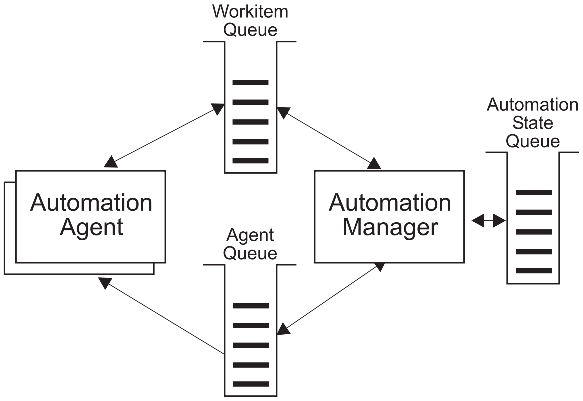


*Figure 4. WebSphere MQ Queues*

The three queues are shared between the PAM and the secondary automation managers (SAMs). When the PAM fails, an SAM becomes the new PAM and takes over the shared queues in a consistent state. No requests are lost because both the actual state of the automated resources, and also the unprocessed requests (work items) – even those that were made during the takeover phase – and all unprocessed orders for the automation agents are known. Only the execution time may be delayed.

When you use WebSphere MQ for manager-agent communication and status backup, it is recommended that you automate WebSphere MQ and let it be started and stopped by SA z/OS. Collecting all the WebSphere MQ and DB2 instances in a basic group allows you to monitor these prerequisites of SA z/OS as you would do with normal application automation.

Automation of WebSphere MQ by SA z/OS implies that WebSphere MQ is not yet available when the automation manager is started. In this situation, the automation manager will communicate with the automation agents through XCF services until it has started its local WebSphere MQ. This phase is called the *startup phase*. As soon as WebSphere MQ is up, the automation manager switches over to WebSphere MQ for communication with the automation agents and for status backup. This means that both the SA z/OS automation manager and automation agents are now WebSphere MQ applications.

Similarly, when the PAM has shut down its local WebSphere MQ, and there is no SAM left for a takeover, the PAM will switch back again to XCF communication. This phase is called the *shutdown phase*.

During these ″WebSphere MQ-less″ phases, resources can change their status, and the information about these changes should be preserved for an eventual successor of the actual PAM. This applies not only to a shutdown and subsequent restart of

SA z/OS, but also to a failure of the PAM and a subsequent takeover by a SAM during the startup phase. To this end, SA z/OS maintains a *takeover file*.

**The Takeover File:** Every status change during the startup and shutdown phase of SA z/OS is recorded in the takeover file. The information in this file is kept consistent by maintaining two compartments for each resource record. These compartments are used alternately to store the changes. This ensures that a consistent and reasonably current version of the resource information exists even when the PAM fails in the middle of an update of the takeover file.

If the PAM fails during startup, a SAM becomes the PAM as before. But now, the new PAM reads the takeover file and starts with the information contained in it. When the shutdown phase is terminated normally or abnormally, the takeover file will be used for a restart of SA z/OS. In this way, a hot takeover or restart is possible even when WebSphere MQ is not available. This of course requires that the takeover file be shared between the PAM and the SAMs.

If a VSAM I/O error occurs during the hot start or takeover, this causes the PAM to initialize with a warm start rather than a hot one. If a VSAM error occurs, you will be notified and asked to choose one of the following options:

- Retry reading the takeover file.
- Continue with a warm start.
- Cancel the hot start and trigger a takeover.

It is also possible that a VSAM I/O error might occur while the PAM is running that causes the PAM to disable recovery of the state information. This now leads to the following behavior:

1. The in-storage state information is not written to the takeover file when the PAM terminates.
2. A WTOR is sent to the operator to enable the takeover file, so that the previous version is used.

Figure 5 shows the timeline for the WebSphere MQ solution when WebSphere MQ is automated by SA z/OS.

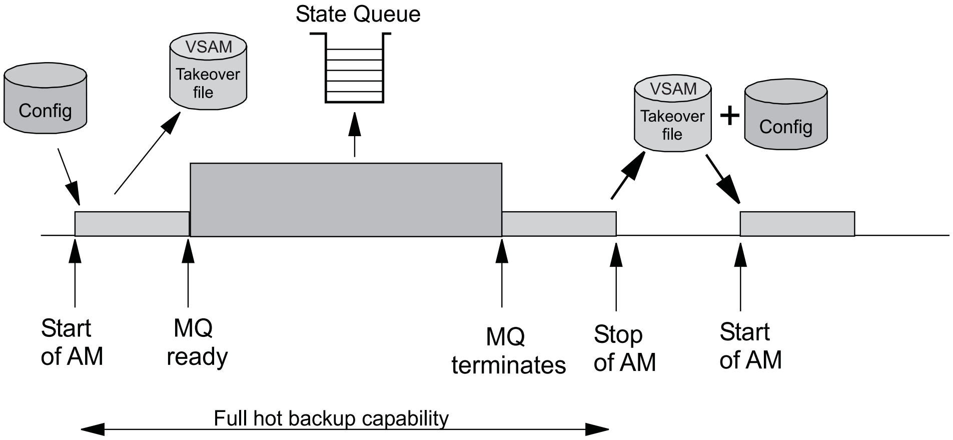


*Figure 5. Automating WebSphere MQ with SA z/OS*

The sequence of events is as follows:

1. During the startup phase, when WebSphere MQ is not yet running, the PAM uses XCF for communication and stores every status change in the takeover file.
2. If the PAM or its system fail during the startup phase, an SAM becomes the new PAM. The new PAM reads the actual state of the automatable resources from the takeover file. (Note that this is not represented in Figure 5.)
3. As soon as WebSphere MQ is up, the PAM switches to WebSphere MQ and writes the status updates into the Automation State Queue.
4. If the PAM or its system fail while WebSphere MQ is running, an SAM becomes the new PAM. The new PAM uses the information in the shared WebSphere MQ queues. (Note that this is not represented in Figure 5.)
5. When the PAM has shut down its own local WebSphere MQ, the reaction of SA z/OS depends on whether an SAM is available:
   - If there are any SAMs, one of these becomes the new PAM. (Note that this is not represented in Figure 5.)
   - If there are no SAMs, the current PAM enters the shutdown phase. It switches back to XCF and the takeover file.
6. During the shutdown phase, the PAM stores every status change in the takeover file. After the PAM has terminated normally or abnormally, the information in the takeover file will be used for a restart of SA z/OS.

**Note:** The takeover file substitutes for the Automation State Queue when WebSphere MQ is not available. There are, however, no corresponding substitutes for the Work Item Queue or the Agent Queue. Therefore, all pending work items and orders to the automation agents will be lost when the PAM fails during the startup or shutdown phase.

For this reason, you should keep the startup phase as short as possible, and define your automation policy so that the local WebSphere MQ manager (local for the PAM) and its associated DB2 are started simultaneously immediately after JES is up.

The following section contains an overview of various recovery scenarios.

**Some Problem Scenarios and How SA z/OS Reacts:** The following are some examples of how SA z/OS reacts when there is a:

**System Breakage with Running Automation Manager and Automation Agent**
A waiting secondary automation manager will automatically take over the responsibility of the failed primary automation manager. Of course the broken automation agent is not moved, because all the broken resources have gone anyway. However the new automation manager will detect the system collapse and react accordingly.

**An Automation Manager Breakage**
A waiting secondary automation manager or the ARM-restarted primary automation manager will automatically take over the responsibility.

**A WebSphere MQ Manager Breakage**
A connected automation agent will wait for the ARM-initiated WebSphere MQ manager restart. A connected automation manager will automatically trigger a takeover in the case of active processes, otherwise this automation manager will also wait. Note that even in the case of a WebSphere MQ manager abend or problem, the automation agent is still able to perform message automation.

**A DB2 Problem or Breakage**

After DB2 first comes up and SA z/OS has been able to access the WebSphere MQ queues for the first time, DB2 is actually no longer needed. Therefore any of these cases can be completely automated even in the full WebSphere MQ-supported fashion.

**A CF Outage**

At this point in time, SA z/OS will automatically restart its automation processing from the existing static configuration (WARM start).

**A Takeover Resulting in the Same Problem**

When the new automation manager detects that a work item has been rolled back twice, the process will be stopped and a WARM initialization will be triggered, thus preventing endless retries failing with the same persistent problem.

**WebSphere MQ Queue Problems**

See "WebSphere MQ Exception Processing" on page 42.

## WebSphere MQ Considerations

This section assumes that you have selected WebSphere MQ for manager-agent communication and status backup.

**Peer Recovery Considerations:** Refer to the WebSphere MQ V5.3 documentation for all aspects of WebSphere MQ sysplexwide peer recovery. Because SA z/OS exploits this technology, you also gain this functionality. The following are important considerations when planning for SA z/OS to be a WebSphere MQ shared queues exploiter.

The basic setup consideration is whether or not you choose to have a dedicated WebSphere MQ QSG (Queue Sharing Group) just for SA z/OS.

Peer recovery requires that a failed WebSphere MQ instance should be restarted using either the z/OS Automatic Restart Manager or SA z/OS itself.

To roll back or complete the broken automation manager activities (*UOW*s in WebSphere MQ terminology), WebSphere MQ can use a different WebSphere MQ manager instance of that QSG (Queue Sharing Group).

SA z/OS will ensure that all pending work has been rolled back before the new primary automation manager starts accessing the queues.

There are no special considerations for DB2 should there be a takeover. DB2 is not involved in the WebSphere MQ peer recovery functions.

Because SA z/OS provides the capability to automate its prerequisites, SA z/OS may be used initially to start WebSphere MQ and DB2. z/OS Automatic Restart Manager or SA z/OS may be used to restart the WebSphere MQ instances, and SA z/OS can be used to observe the status of its prerequisites as well as finally to stop it.

Collecting all the WebSphere MQ and DB2 instances in a basic group allows you to monitor all SA z/OS prerequisites as you would do with normal application automation.

**WebSphere MQ Exception Processing:** WebSphere MQ services may fail. Assuming that the WebSphere MQ setup and queue definitions are correct, there is still a chance of running into a WebSphere MQ exception. Basically, these exceptions can be categorized into:

- **Recoverable errors —** these can be recovered by running the failed WebSphere MQ service again after a certain time.
- **Unrecoverable errors —** these cannot be recovered automatically.

The automation agents will react to unrecoverable exceptions by disconnecting from either the WebSphere MQ manager or WebSphere MQ queue. However the sysplex communication task will not be stopped. It will continuously try to re-establish the broken connection.

The automation manager will trigger a takeover if there is a local WebSphere MQ manager problem with active transactions.

Problems with the Automation State Queue are handled differently. A takeover should be avoided if possible, because it probably cannot be successfully completed. For cases where this is possible the queue will be closed and GET/PUT disabled. Message INGY1107 is then issued. Processing on the automation manager continues because the data is still in storage. You then have the chance to repair the queue, for example by:

- Redefining the queue on a different CF
- Redefining the queue on a different CF Structure
- Increasing the maximum number of possible messages.

The current automation manager will continuously monitor whether there is a new Automation State Queue that has both GET and PUT reenabled. If this automation manager finds a queue with this attribute it will try to reinstall the queue. Having done this, SA z/OS is fully recoverable again.

**Queue Full Considerations:** WebSphere MQ queues can become full. No further MQPUTs are possible unless some MQGETs remove messages.

The current active SA z/OS automation manager automatically performs the recovery from a queue full condition. It can be considered as a recoverable exception as described above.

Situations where a queue full condition can occur are:

1. For the Work Item Queue:
   - An automation manager is not available to pick up the work item requests (for example, it has just stopped or is restarting).
   - An automation agent-based automation CLIST repetitively sends requests to the automation manager. Because these automation manager requests can be generated by an automation program, this program may loop.
2. For the Agent Queue:
   - Automation agents are not available or able to process orders or responses in time.
   - Many concurrent large response blocks.
3. For the Automation State Queue:
   - Dynamic Configuration Reloads drastically increase the amount of Automation State information.

The samples delivered with SA z/OS combined with a healthy system should not result in such a situation. It is more an indication that something is wrong but SA z/OS tries its best to survive.

The recovery action for an Automation State Queue full condition is described in "WebSphere MQ Exception Processing" on page 42.

For the Automation Work Item Queue and the Automation Agent Queue, three additional message counters have been introduced to act as thresholds:

- **Low Threshold —** if the number of messages is below this limit, operations on that queue are fine, and no recovery actions are taken.
- **High Threshold —** if the number of current active messages reaches this count, recovery actions are taken. To stop this recovery mode again, the number of messages must fall below the low threshold counter.
- **Max_Queue_Depth —** at this time, further MQPUTs are rejected, however SA z/OS would retry.

To see these thresholds, use the INGAMS command that is described in *IBM Tivoli System Automation for z/OS Operator's Commands*.

# Chapter 4. Planning to Install TEC Notification by SA z/OS

This section contains information required for the installation of TEC Notification by SA z/OS.

## Introduction of TEC Notification by SA z/OS

SA z/OS notification can be used to notify Tivoli Enterprise Console® (TEC) about an automation problem on z/OS by sending an event to the TEC event server.

For this purpose, on z/OS systems, messages or alerts are transformed into Tivoli events and sent to the *TEC event server* that is running on a Tivoli-managed node in your network.

These events in turn may:

- Cause a notification of a Tivoli administrator on the TEC
- Be correlated with other events on the TEC event server
- Result in opening a trouble ticket, for example, dependent on what you programmed at the TEC event server

TEC event server was introduced as a new notification target for SA z/OS. Note that only those messages that indicate critical situations and alerts are forwarded as TEC events to the TEC event server using the appropriate NetView Event/Automation Service Adapter.

A Tivoli administrator who wants to deal with a problem indicated by an event that is forwarded to the TEC event server by SA z/OS needs access to the affected z/OS system. You may use the Tivoli NetView 3270 Management Console for this. With TEC Notification by SA z/OS, the TEC administrator may log on to the NetView operator console by starting the NetView 3270 Management Console from the TEC console by executing a task. See *IBM Tivoli System Automation for z/OS User's Guide* for a description of the graphical interface on how to achieve this.

**Note:** Forwarding of SA z/OS messages to TEC will not start until SA z/OS and the Event/Automation Service are up and running. SA z/OS messages issued during SA z/OS startup will not be forwarded to TEC.

## Environment Configurations

Several products are involved in TEC Notification by SA z/OS:

- Tivoli NetView for z/OS
- Tivoli NetView Event/Automation Service
- Tivoli Enterprise Console (TEC)

You can run TEC Notification by SA z/OS in two configurations:

- **Local Configuration**: The message adapter or alert adapter is running on the same z/OS system on which SA z/OS is also running. The adapters are local to the SA z/OS which is issuing and forwarding messages and alerts to the Tivoli Enterprise Console. Such a configuration for message forwarding is illustrated in Figure 6 on page 46.

- **Distributed Configuration**: The message adapter or alert adapter is running on an z/OS system different from the one on which SA z/OS is running and issuing messages and alerts. In this scenario, the z/OS system running the adapters must be the SA z/OS automation focal point system. Such a configuration for message forwarding is illustrated in Figure 7 on page 47.

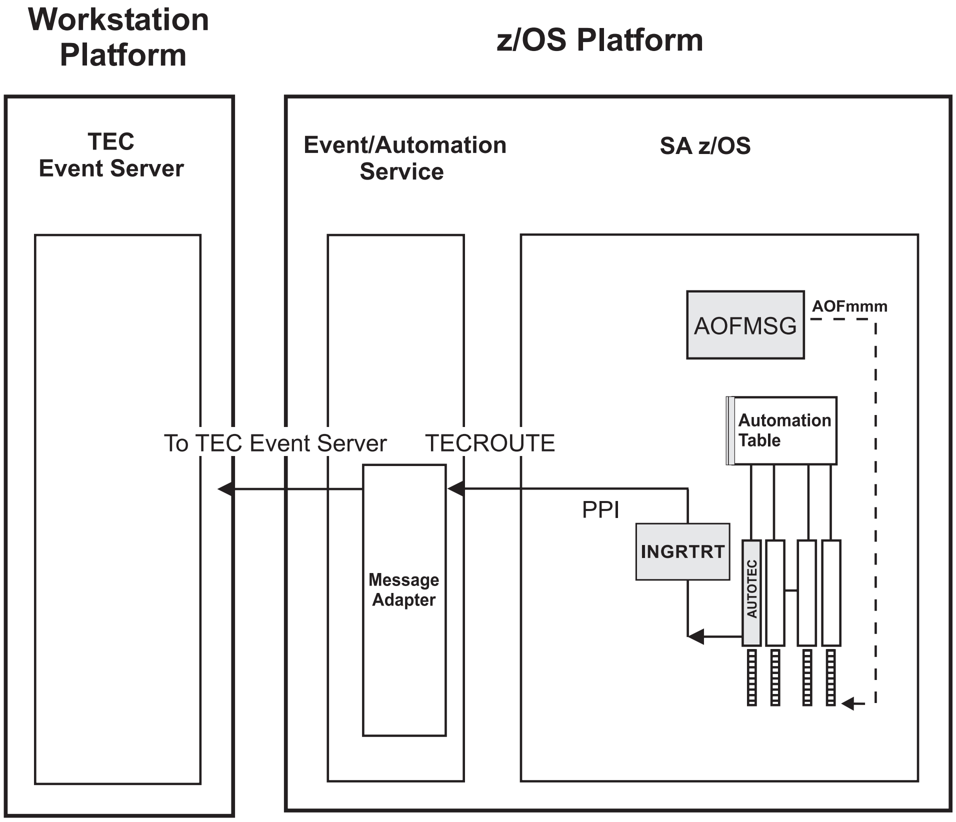


*Figure 6. Local Configuration: NetView Event/Automation Service Local to the SA z/OS Source of Messages*

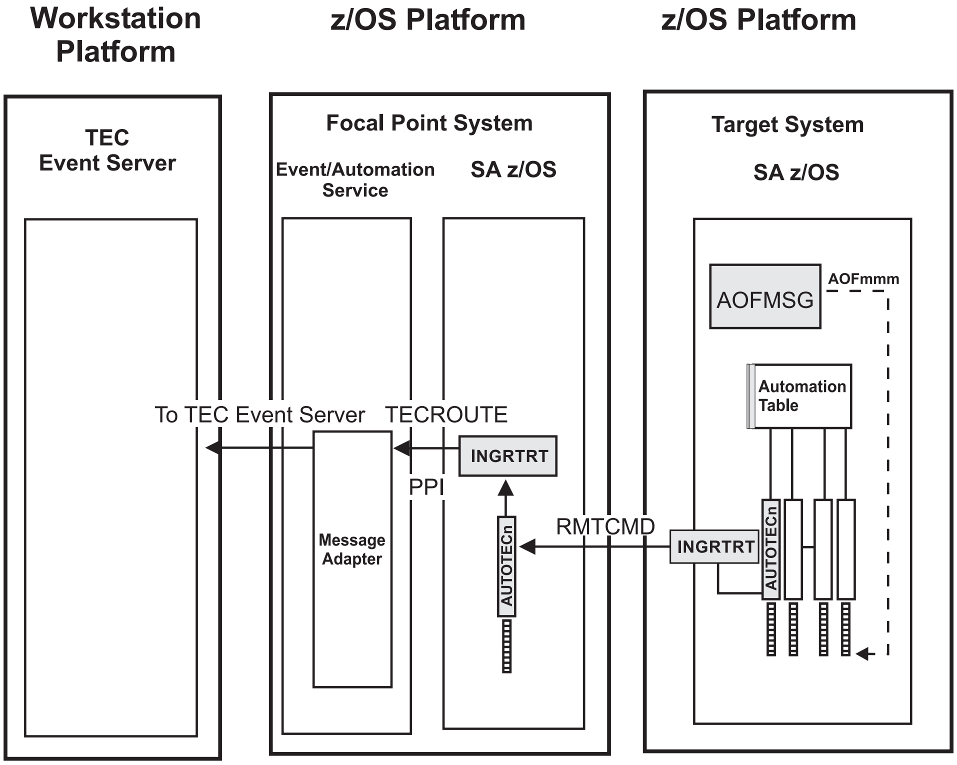


*Figure 7. Distributed Configuration: NetView Event/Automation Service Remote to the SA z/OS Source of Messages*

Section "Environment Configurations" on page 45 describes the two configurations in which you can run TEC Notification by SA z/OS. How to install this feature is described in the following sections:

- "Installing and Customizing the TEC Event Server Workstation" on page 164
- "Activating the Installed Files" on page 165

The customization part comprises the following steps:

- "Step 15: Customization of NetView for TEC Notification by SA z/OS" on page 119 describes how to customize your SA z/OS and TEC Notification by SA z/OS installations on the z/OS system for both the local and distributed configuration as described in "Environment Configurations" on page 45.
- "Installing and Customizing the TEC Event Server Workstation" on page 164 describes how to install and activate the workstation code on the TEC Event Server.

You can find more conceptual information about TEC Notification by SA z/OS and information on how to use it in *IBM Tivoli System Automation for z/OS User's Guide*.

# Chapter 5. Planning for the NMC Environment

The information in this section helps you to plan the configuration of the components in your NMC environment.

## NMC Exploitation Topology

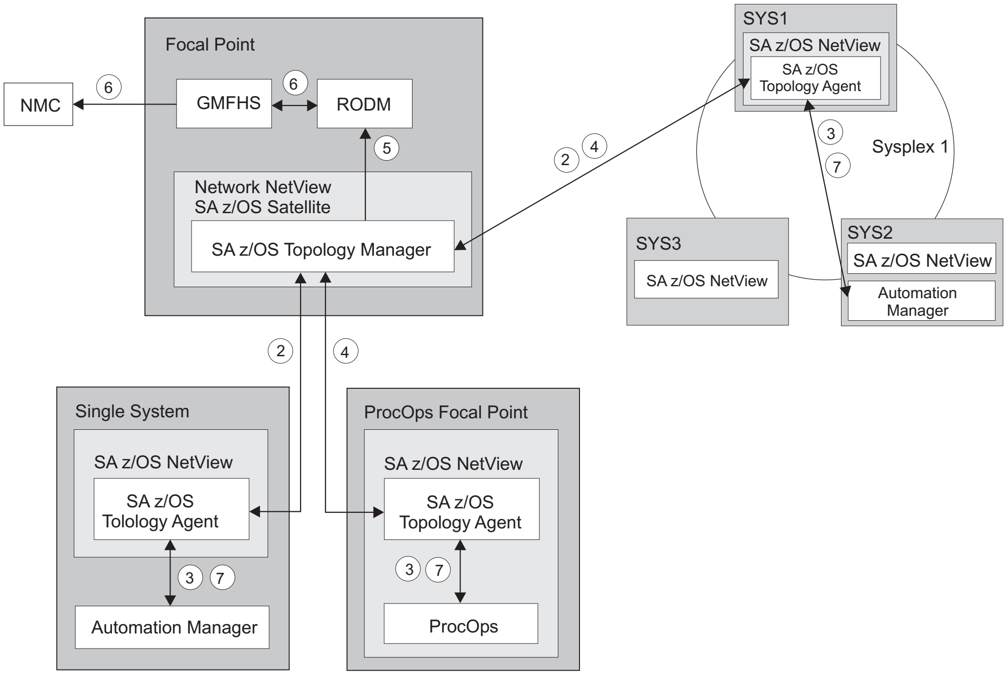


*Figure 8. The SA z/OS Environment for NMC Support*

Figure 8 shows how in a SA z/OS configuration the involved components communicate to produce graphical output information:

1. At initialization time, the SA z/OS topology manager knows the target systems for automation.
2. The SA z/OS topology manager contacts the SA z/OS topology agents on all sysplexes or stand-alone systems or, for processor operations, it contacts the processor operations focal point to obtain the required information.
3. The SA z/OS topology agents contact the related automation managers or the processor operations component respectively to find out the status from the systems and resources.
4. Then the SA z/OS topology agents report this information to the SA z/OS topology manager on the focal point.
5. The SA z/OS topology manager feeds the RODM data base with the achieved information.
6. The NMC workstation on the operator's request can retrieve the RODM data to produce the defined views.
7. Also, at initialization time, the automation managers get the order to inform the related SA z/OS topology agents whenever status changes occur. Then the

SA z/OS topology agents will route the status change information to the SA z/OS topology manager which will update the RODM data base.

## Planning to Install the NMC Workstation

Make sure that you have a working NMC environment with the required functions (for example, RODM, GMFHS, NMC Topology Server, NMC Topology Console, NMC 3270 Management console), as part of your NetView installation available.

For information on how to install the NMC, refer to *Tivoli NetView for z/OS Installation: Configuring Graphical Components* and *NetView Management Console User's Guide*. The information about what to do to enable your NMC environment installation for use in SA z/OS is described in "Installing the NMC Workstation" on page 157.

If you plan to use Kanji support for NMC keep in mind that all the NetView workstations in the domain must support the character set you decide to use. Multilingual support is not available.

## Running Multiple NetViews

If you use two NetViews and you want to monitor resources using the NMC workstation, bear in mind that the NMC workstation must be linked to NetView Graphic Monitor Facility Host Subsystem (GMFHS) on the Networking NetView which has a connection to RODM. See Figure 10 on page 51. You can operate network and SA z/OS resources via RODM and have SA z/OS running in another NetView to control the automation resources. This, however, requires a subset of SA z/OS, referred to as the SA z/OS satellite, to be installed on the Networking NetView. See "Step 25: Install an SA z/OS Satellite" on page 134 for details.

If you run the Networking Automation NetView only on the focal point, you cannot have your resources automated by SA z/OS.

If you run the System Automation NetView only on the focal point, you cannot have networking resources in RODM, but only SA z/OS resources that you automate.

Alternatively, you can run both the Networking Automation and the System Automation on the same NetView. This way, you can save storage and CPU costs because of the reduction in the duplication of, for example, tasks and logs. But more important, it reduces maintenance and system programmer costs. See Figure 9 on page 51 for details.

In such an environment all functions are handled by that NetView. You may want to give the individual NetView tasks different priorities, for example, the System Automation tasks need to run above the VTAM's priority, whereas others (Networking Automation) need to run at a lower priority. This is achieved with z/OS Workload Manager Enclaves support.

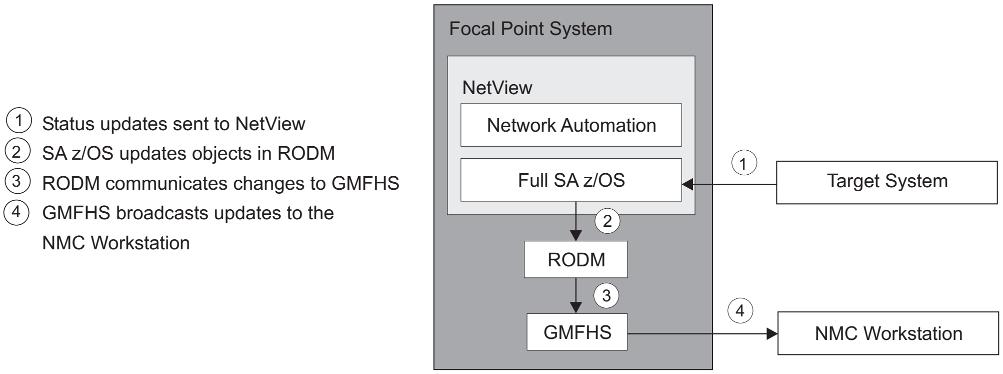

1 Status updates sent to NetView
2 SA z/OS updates objects in RODM
3 RODM communicates changes to GMFHS
4 GMFHS broadcasts updates to the NMC Workstation
Focal Point System
NetView
Network Automation
Full SA z/OS
RODM
GMFHS
1
2
3
4
Target System
NMC Workstation

*Figure 9. SA z/OS Enterprise with Networking Automation and System Automation running on the same NetView*

Figure 10 illustrates the flow of data from a target system to the focal point when two NetViews are used on the focal point: one for Networking Automation and one for System Automation.

1. The target system data is sent to the Networking NetView at the focal point via Command Handler or Alerts; the AAO AOFSENDALERT will dictate which forwarding mechanism is used. (Alerts from processor operations are sent directly to the Automation NetView).
2. The satellite z/OS automation (focal point) receives the data that is sent from the targets and updates objects in RODM appropriately.
3. NetView Graphic Monitor Facility Host Subsystem (GMFHS) becomes aware of status updates.
4. GMFHS broadcasts updates to the operator workstation.

When an operator initiates a command or routine from a workstation, the action flows back to the Networking NetView for processing in the reverse direction from that shown in Figure 10.

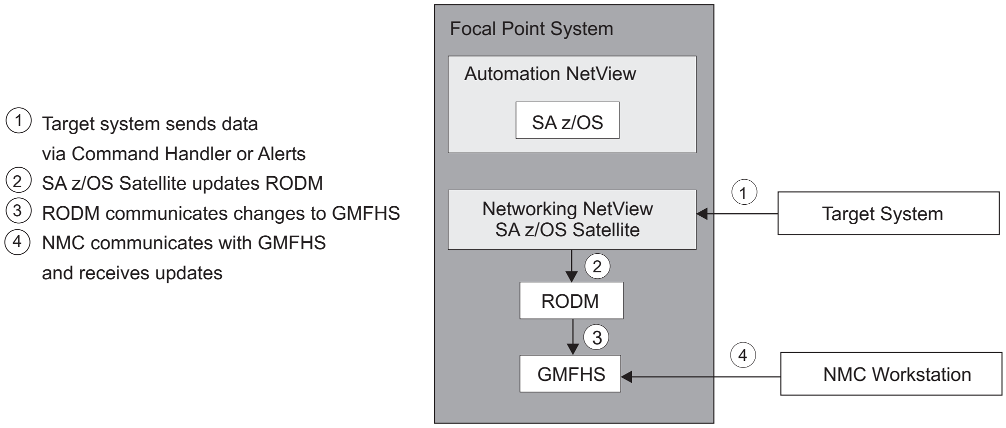

1 Target system sends data via Command Handler or Alerts
2 SA z/OS Satellite updates RODM
3 RODM communicates changes to GMFHS
4 NMC communicates with GMFHS and receives updates
Focal Point System
Automation NetView
SA z/OS
Networking NetView
SA z/OS Satellite
RODM
GMFHS
1
2
3
4
Target System
NMC Workstation

*Figure 10. SA z/OS Enterprise Using a Networking NetView and an Automation NetView*

# Chapter 6. Planning for Automation Connectivity

This chapter provides background on SA z/OS. It includes what a focal point system is and what targets are, and how to define a network of interconnected systems, known as an *automation network*, to SA z/OS for purposes of monitoring and controlling the systems. The procedures and examples in this chapter assume that VTAM definitions for systems in the automation network are in place and available as input.

## The Focal Point System and Its Target Systems

SA z/OS allows you to centralize the customization, monitoring, and control functions of the multiple systems or images that make up your enterprise using a single, centrally located z/OS system. This controlling z/OS system is called the focal point system. The systems it controls are called target systems. These systems communicate using XCF, WebSphere MQ and NetView facilities.

## Defining System Operations Connectivity

This section discusses the following aspects of defining system operations connectivity:
- "Multiple NetViews"
- "Overview of Paths and Sessions"

### Multiple NetViews

| The number of NetViews that run in your SA z/OS complex affects how you plan
| for it. SA z/OS can operate with just one NetView at its focal point. It is your
| decision whether you want to run the *Networking Automation* and the *System*
| *Automation* on separate NetViews.

### Overview of Paths and Sessions

This section provides an overview of the following:
- "Message Forwarding Path" on page 54
- "Gateway Sessions" on page 54

## Message Forwarding Path

SA z/OS generates and uses messages about significant actions that it detects or takes such as a resource status change. In addition to sending these messages to operators on the same system, SA z/OS can forward them from target systems to a focal point system and can route commands and responses between systems, using a message forwarding path. This path is defined in your policy. Key components in a message forwarding path include:

- A primary focal point system
- A backup focal point system
- A target system or systems
- Gateway sessions connecting systems. Gateway sessions use inbound and outbound gateway autotasks. Communication is via the NetView RMTCMD or XCF when the focal point system and target system are in the same sysplex.

Using a message forwarding path, a focal point system can monitor several target systems.

SA z/OS uses notification messages to update the status of resources displayed on the status display facility (SDF). Routing notification messages over the message forwarding path helps consolidate monitoring operations for multiple systems on the SDF at a focal point system. See *IBM Tivoli System Automation for z/OS User's Guide* for details on configuring SDF for a focal point system-target system configuration.

## Gateway Sessions

**Outbound and Inbound Gateway Autotasks:** Each gateway session consists of:

- Two gateway autotasks on each system:
  - One autotask for handling information outbound from a system, called the outbound gateway autotask. This establishes and maintains all connections to other systems. It sends messages, commands, and responses to one or more systems.
  - One autotask for handling information incoming from another system, called the inbound gateway autotask. A system can have one or more inbound gateway autotasks, depending on the number of systems to which it is connected.

Figure 11 shows a single gateway between two SA z/OS agents, IPUNA and IPUNB.

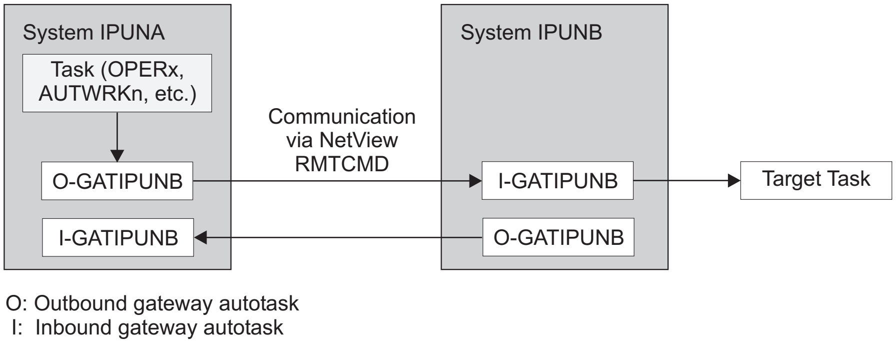

System IPUNA
Task (OPERx, AUTWRKn, etc.)
O-GATIPUNB
I-GATIPUNB
Communication via NetView RMTCMD
System IPUNB
I-GATIPUNB
O-GATIPUNB
Target Task
O: Outbound gateway autotask
I: Inbound gateway autotask

*Figure 11. Single Gateway Example*

There is one task handling all outbound data. This task is set up at SA z/OS initialization time. Normally the task has a name that begins with GAT and ends with the domain name. So for IPUNA, the gateway task is GATIPUNA.

When VTAM becomes active, the gateway task (GATOPER) issues a CONNECT call to the remote system, IPUNB in our example. If the GATIPUNA task on the remote system is not already active, it will be started automatically by NetView.

All requests initiated by system IPUNA and destined for system IPUNB use the task pair GATIPUNA. Likewise all requests that originate on system IPUNB and are destined for system IPUNA use the pair GATIPUNB. In other words the communication is half-duplex. There is one task pair responsible for the outbound traffic while another task pair is in charge of the inbound traffic. Each pair consists of a sender - running on the local system and receiver that runs on the remote system.

Disallowing the starting of the receiver task protects the local system from getting requests from the remote system.

The task structure is similar when using XCF as the communication vehicle. Using the "GATxxxx" task as the receiving and processing task on the remote side gives a dedicated task pair for the communication between the two systems. This task pair exists twice, once for each outbound communication. It is important to notice that the standard RPCOPER is not used for the processing of the remote procedure call.

In the automation policy for each system in an automation network, you need to define only the outbound gateway autotask (see *IBM Tivoli System Automation for z/OS Defining Automation Policy*). However, in the NetView DSIPARM data set member DSIOPF, you must define all gateway autotasks, both inbound to and outbound from a system, as operators.

You define the outbound gateway autotask by defining the GATOPER policy item for the Auto Operators policy object in the customization dialog. You must specify an operator ID associated with the GATOPER function in the Primary field on the Automation Operator NetView panel. See *IBM Tivoli System Automation for z/OS Defining Automation Policy* for more information.

For this example, the operator ID for the system CHI01 outbound gateway autotask is GATCHI01. Similarly, any operator ID for an inbound gateway autotask is the prefix GAT combined with the inbound gateway domain name.

Figure 12 on page 56 shows three systems: CHI01, ATL01, and ATL02. System CHI01 is the focal point for forwarding messages from target systems ATL01 and ATL02. In Figure 12 on page 56, gateways are designated as follows:

**O** Outbound gateway autotask

**I** Inbound gateway autotask.

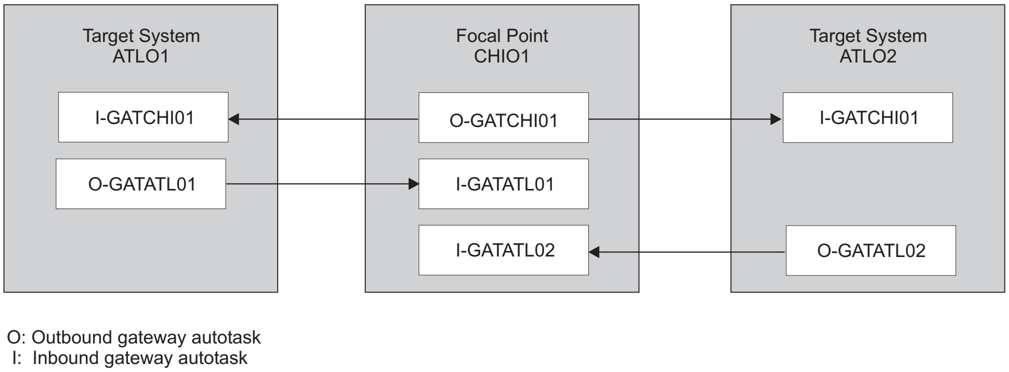


*Figure 12. Example Gateways*

**How Gateway Autotasks Are Started:** Gateway autotasks establish a connection between systems when any system receives the following NetView message:

```
DSI112I NCCF READY FOR LOGON AND SYSTEM OPERATOR COMMANDS
```

When this message is received, the following steps occur:

1. The outbound gateway autotask tries to establish an outbound session with the remote system.
2. A gateway session between two systems is established when the outbound gateway autotask has established its outbound session to the remote system.

This process automatically establishes outbound and inbound connections for systems without human operator intervention.

**How Gateway Sessions Are Monitored:** Optionally, gateway sessions can be monitored by a command executed periodically. The time interval is set in the field Gateway Monitor Time. in the Automation Setup policy item for the System policy object.

See *IBM Tivoli System Automation for z/OS Defining Automation Policy* for details. The ID of the timer created to monitor gateway sessions is AOFGATE. This timer will not be set if NONE is entered for Gateway Monitor Time.

If SA z/OS detects that any gateway session is inactive during the monitoring cycle, it tries to restart the session.

## Automatically Initiated Terminal Access Facility (TAF) Fullscreen Sessions

Using the FULL SESSIONS policy item of the Network policy object, you can set up automatically-initiated terminal access facility (TAF) fullscreen sessions from within SA z/OS. *IBM Tivoli System Automation for z/OS Defining Automation Policy* describes how to define applications with which SA z/OS operators can establish TAF sessions automatically using the SA z/OS NetView interface.

## Using Focal Point Services

Once an automation network is configured, you can use the message forwarding path to route messages, commands, and responses between systems. SA z/OS operators can display the status of gateway autotasks and TAF fullscreen sessions using the SA z/OS operator commands. Details on these operator activities are in *IBM Tivoli System Automation for z/OS User's Guide*.

# Defining Processor Operations Communications Links

After determining that you plan to use the processor operations functions, you must decide the type of communication link from your focal point system to your support element. Processor operations supports the following types of communication connections:

- NVC
- SNMP
- TCP/IP

## Meeting Availability Requirements

In order to reduce the interruption time in case of processor operations communication problems, the following facilities are available:

- Backup Support Element
- Alternate focal point system

### Backup Support Element

Selected types of the CMOS-S/390 processor family and all zSeries processors have a second Support Element installed, operating in hot-standby mode. If the primary Support Element fails, the backup SE is automatically activated as the new primary Support Element. The SE configuration information is always duplicated, so the new primary SE has the same configuration information as the failing one including the SNA or IP network addresses.

### Alternate Focal Point System

An alternate focal point system can be used, in addition to the primary focal point system, to minimize the effect of a focal point system outage. If a focal point system must remain operational all the time, an alternate focal point system can be operated in a take-over mode.

### Alternate Focal Point for SNA based NVC connections

If you plan to install an alternate focal point system, you must include one or more 37xx communications controllers. Each controller must be equipped with a channel adapter. The Network Control Program (NCP) must be installed in the communications controllers. You can use a 3174 subsystem control unit in place of the 37xx.

However, the alternate focal point system operator is not automatically notified of the loss of the session between a focal point system and a NetView connection. This notification is instead received by the operator of the failed focal point system, which is the primary focal point system.

### Alternate Focal Point for SNMP connections

If you plan to use a second focal point system for your processor operations SNMP connections, make sure that the TCP/IP USS stack is always up and that your IP network allows the communication between the alternate focal point and the Support Elements.

### BCP internal interface considerations

If you have customized SA z/OS to use the BCP internal interface for the sysplex hardware automation, each system being a member of the sysplex has its processor hardware connection activated and can issue hardware requests to the SEs of the other sysplex members. The SA z/OS internal code routes the supported hardware commands only to a system in the sysplex with a functioning hardware interface to make sure the request can be processed successfully.

## Task Structure for Processor Operations

For processor operations there is a task structure that is modular; distinct types of SA z/OS tasks handle different work assignments. The types of SA z/OS tasks are:

- Target control tasks
- Message monitor tasks (used for SNMP and TCP/IP connections only)
- Recovery task
- Start task
- Polling task
- OCF-CI task

SA z/OS allows up to 999 tasks of each of the first three types, but only one recovery task and one processor operations start task. Because SA z/OS tasks are z/OS tasks that require system services and also add to the load running in the NetView address space, you should only define as many tasks as are needed.

The following guidelines help you match the number of SA z/OS tasks to your SA z/OS configuration.

- The number of message monitoring tasks for target systems connected with a SNMP connection should be identical to the number of target control tasks in your environment.
- The number of target control tasks should be less than or equal to the number of target hardware defined. If you plan to use the processor operations group and subgroup support for the common commands, the total number of target control tasks should be equal to the number of concurrently active target hardware systems.
- In consideration of focal point performance, limit the total number of tasks to a number your system can handle.

### Target Control Tasks

The number of target control tasks is automatically calculated and set.

Target control tasks process commands. A target system is assigned to a target control task when the target system is initialized. More than one target system can be assigned to the same target control task. A target control task is a NetView autotask.

### Message Monitor Tasks

**Note:**
When you are using a NetView connection, these tasks are not required.

The number of message monitor tasks is automatically calculated and set.

Message monitor tasks receive SNMP traps from the Support Element's SNMP clients and receive messages from the PSMs and their associated VM second level systems at the focal point system. The traps and messages are broadcast to the appropriate tasks and operators.

### Recovery, Start, and Polling Tasks

Automation for resource control messages runs under the recovery task, which is a NetView autotask. Processor operations also uses the recovery task for processing of recovery automation commands. Normally, this task is idle. It is generated automatically when you generate NetView autotask definitions from the configuration dialogs.

The startup task, a NetView task, is used to establish the processor operations environment with the NetView program and to start the other NetView tasks needed for processor operations to function. The startup task is only active during processor operations start (ISQSTART).

The polling task, another NetView task, is used to poll the processors using NetView connections. You determine both the polling frequency and polling retries to be attempted. (These polling functions are specified using the NetView connection path definition panels in the configuration dialogs.) This task is generated automatically when you generate the NetView Autotask definitions from the customization dialogs. This NetView task enables SA z/OS to verify and update operations command facility-based processor status.

### Processor Operations OCF-CI Task

The OCF-CI task receives messages sent to the support element on the console integration interface. These messages come from a target OCF-based or parallel enterprise server operating system. The task broadcasts these messages to the appropriate processor operations task and operators interested in processor operations. For information about interested operators, see *IBM Tivoli System Automation for z/OS User's Guide*. The OCF-CI task, a NetView autotask, is singular. Only one is required, and it is required only for processors connected with a NetView connection. This task is not required for SNMP connections.

# Planning Processor Operations Connections

This section describes making the hardware connections. It is divided into subsections for each set of hardware connections:

- "Preparing the Processor Operations Focal Point System Connections" and "Preparing the Alternate Focal Point System Connections" on page 60 for focal point system connections
- "Preparing the Target System Connections" on page 61 for target system connections. This section also discusses complex connection configurations.

## Preparing the Processor Operations Focal Point System Connections

The physical path for the focal point system consists of connections from the HMC, SE, or PSM to the focal point system. SA z/OS processor operations supports the following types of communication connections:

- NVC
- SNMP
- TCP/IP

## TCP/IP Firewall-Related Information

The TCP/IP SNMP connections of ProcOps use port number 3161. This is the port number that Support Elements or Hardware Management Consoles use to communicate with SA z/OS ProcOps or other applications using the z900 API.

In case you have firewalls installed between the processor LAN and the LAN that SA z/OS ProcOps belongs to , make sure port 3161 is registered to prevent SE/HMC responses from being rejected.

## Preparing the Alternate Focal Point System Connections

An alternate focal point system can be connected to your DP enterprise in addition to the primary focal point system.

The physical connection path for the alternate focal point system is identical to that for the primary focal point system. As with the primary focal point system, SA z/OS processor operations supports the following types of communication connections:

- NVC
- SNMP
- TCP/IP

## Connection Example

Figure 13 on page 61 shows an alternate focal point system as well as a primary focal point system connected from an IP or SNA network to the processor hardware LAN.

For SNA networks, an SNA gateway device such as 2216 or 37xx network controller must be connected to the processor LAN. In an SNA network, the NetView connection type NVC can be used. The NVC connection also requires that the processor LAN is a token-ring LAN. Note, that not all processors of the zSeries processor family may support NVC connections.

For an SNMP connection, the processor hardware LAN can be either Ethernet or token ring. With SNMP, a connection can be established either to the Support Element of a CPC, or to an HMC. This HMC must have the CPCs defined you want to manage. This option is not available for SNA based NVC connections.

With TCPIP, a connection can be established to a ProcOps Service Machine on a VM host (PSM).

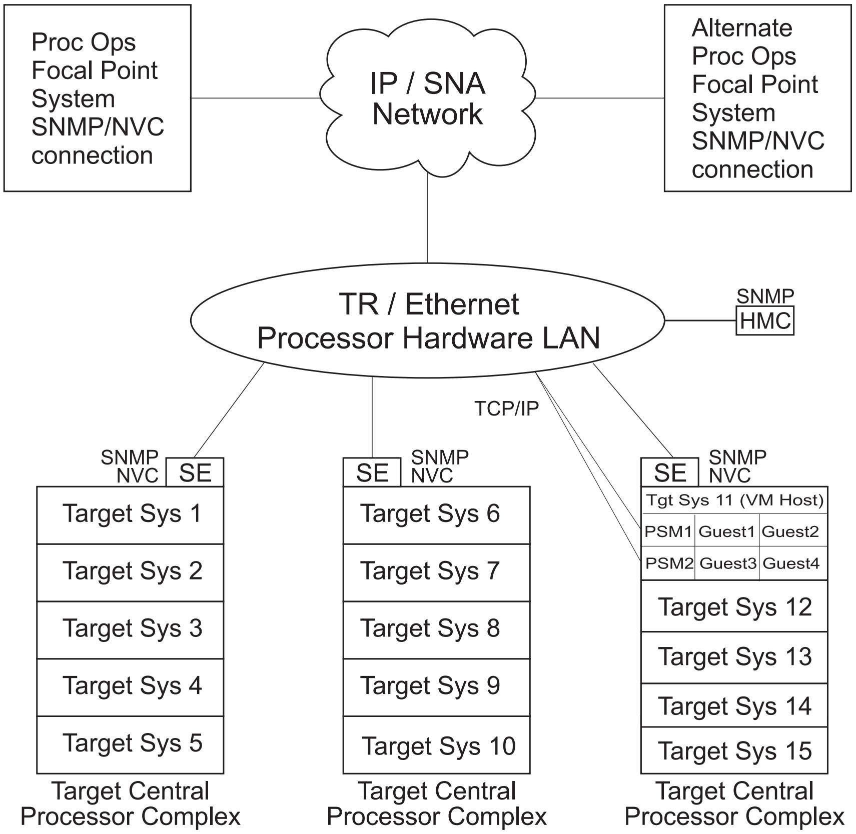


*Figure 13. Alternate and Primary Focal Point System Connections from an IP or SNA Network to the Processor Hardware LAN*

## Preparing the Target System Connections

The supported processor hardware allows you to use the attached Support Element or an HMC (SNMP connections only), connected to the processor hardware LAN for hardware operations management tasks and for operating system control. The Console Integration (CI) function of the SE or HMC is used by processor operations to send commands to an operating system and to receive messages from an operating system. The Operations Command Facility (OCF) of the SE or HMC is used to perform tasks like SYSTEM RESET, LOAD, or ACTIVE.

The usage of CI by processor operations is intended to automate system initialization and recovery tasks. For day-to-day console operation tasks, processor operations CI usage should supplement the operating system command routing facilities of SA z/OS or the available console devices like the 2074 control units.

## Defining I/O Operations Communications Links

When you use SA z/OS on one system to make an operational change to an I/O resource, like a shared ESCON Director, it coordinates the change with other copies of SA z/OS on other systems. This is especially important when the result of the action you are taking removes connectivity - disables I/O paths - so that the systems do not lose access to critical resources. Each copy of SA z/OS interacts with its local system image (for example, via VARY) so the operating system has the chance to ″vote″ on the changes. When one system fails VARYs, SA z/OS takes that as a vote of ″no″ and fails the operation. The copy of SA z/OS from which you initiated the operation then interacts with the other copies on the affected system images to back out VARYs that were successful.

The copies of SA z/OS across your systems also use the network to share information with each other on changes to the I/O configuration and to provide displays that collect I/O information from multiple systems.

To do this, the SA z/OS I/O operations functions on each system image need to intercommunicate. They do this by establishing VTAM sessions between each other. All systems that share access to a given ESCON Director should run SA z/OS to provide the protection described above. Those copies of SA z/OS that do share access to a Director automatically discover each other and establish sessions each time they start.

You can also use the Reset Host function of I/O operations to force two copies of SA z/OS that do not share any ESCON Directors to establish communications. This is useful if you want to benefit from the I/O operations multisystem I/O graphic displays or use its multisystem version of Remove CHP, Restore CHP, Remove Device, or Restore Device, even across system images that don’t use ESCON Directors or have no reason to share them.

I/O operations is able to interact with systems that are running ESCON Manager. I/O operations can interact with VM systems that run ESCON Manager 1.2 to support switching operations (for example, blocking ports or writing an entire saved switch configuration) and for Remove CHP and Restore CHP. I/O operations can interact with z/OS systems that run ESCON Manager 1.3 to support the same operations as for VM, and also the same level of multisystem Query and graphic display requests that ESCON Manager 1.3 itself supports.

To plan for this function, you must review the I/O configuration across the systems that you will define as an enterprise in SA z/OS. You should plan to include in one enterprise all system images that share a given ESCON Director, in order to benefit from the I/O operations configuration change protection and displays.

To enable the VTAM sessions, you must create VTAM definitions as described in “Step 18B: Perform VTAM Definitions” on page 125 to support communications between I/O operations defined as a VTAM application in each of them.

Where images do not automatically use those definitions to start sessions, because they do not share ESCON Directors, you should plan local procedures to use the SA z/OS Reset Host function to force I/O operations to start the sessions.

# Chapter 7. Naming Conventions

## SA z/OS System Names

The information in this section describes name requirements for z/OS systems and for processor operations functions.

All system names defined with the customization dialog in one policy database must be unique.

If your system names currently contradict this restriction, you must change the names before using SA z/OS.

System names defined in the customization dialog for z/OS, VM, TPF, or LINUX systems can have up to 20 characters and must be unique within the SA z/OS enterprise.

When you name elements of your SA z/OS processor operations, use a logical format to create names that are clear to the people using them. The following names can consist of 1 to 8 alphanumeric characters (A-Z,a-z,0-9,#,$,@), cannot contain blanks, and must begin with an alphabetic character:

- Processor or target hardware names
- Target system names
- Focal point name

Processor or target hardware system names, target system names, group names for target systems, and subgroup names for target systems must all be different from one another. Target system names must also be different from processor operations names. For any given system, however, its system name can equal its own processor operations name.

Group and subgroup names for target systems can consist of up to 20 alphameric characters.

Sysplex group names should not be more than 8 characters in length because they are used to address the sysplex or subplex.

## Cloning on z/OS Systems

The SA z/OS cloning capability allows you to specify up to 36 clone IDs to identify a system and to identify an application. These clone IDs are then used to qualify the application job name to ensure a unique job name for each system. The names given to each of these clones must be unique. The z/OS system symbolics and the NetView &domain. variable can also be used.

# Further Processor Operations Names

Image, Load, and Reset profile names are defined at the support element of an OCF-based target processor. They must consist of the characters A-Z and 0-9. Secondary OCF and Image profile names can be up to eight characters; Reset and Load profile names can be up to sixteen characters.

# ESCON Director Ports

This section offers some suggestions for naming ESCON Director ports (dynamic switch ports) and fully utilizing these names in I/O operations display and connectivity commands.

## Reasons for Naming Switch Ports

Assigning names to switch ports:

- Provides an indication of what is on that port. For example, CP01.SYSA.CHP38 indicates that this port is physically connected to processor CP01, on system SYSA, on CHPID 38.
- Allows you, when issuing I/O operations connectivity commands, to refer to ports by name. For example, `BLOCK 3490.46233.CU1.E *` blocks the port connected to interface E of control unit side 01, on the 3490 control unit with serial number 46233. See “Using Port Logical Names” on page 65.
- Allows you, when issuing I/O operations connectivity commands, to change connectivity of an entire system to a control unit. For example, `PROHIBIT CP01.SYSA* 3990.35182* *` removes connectivity from all ports on system SYSA of processor CP01, from all ports on the 3990 control unit with serial number 35182. See “Using Generic Logical Names” on page 66.

## Suggestions for Naming ESCON Director Ports

When naming ports, you should choose names that help identify what the port is connected to. This simplifies the task of entering commands when connectivity changes are required. Following are some suggestions for naming CHPID ports and control unit ports, followed by a figure displaying those ports in an actual configuration.

### Naming CHPID Ports

Name the CHPID ports with three parts: the processor name, followed by the system image name, followed by the CHPID number. For example:

```
CP02.SYSC.CHP40
```

is the port name associated with CHPID 40, on system SYSC of processor CP02.

### Naming Control Unit Ports

Name the control unit ports with four parts: the device type, followed by the serial number, followed by the storage cluster (or control unit side), followed by the interface letter. For example:

```
3990.35182.SC1.E
```

is the port name associated with the 3990 with serial number 35182, on storage cluster 1, interface E.

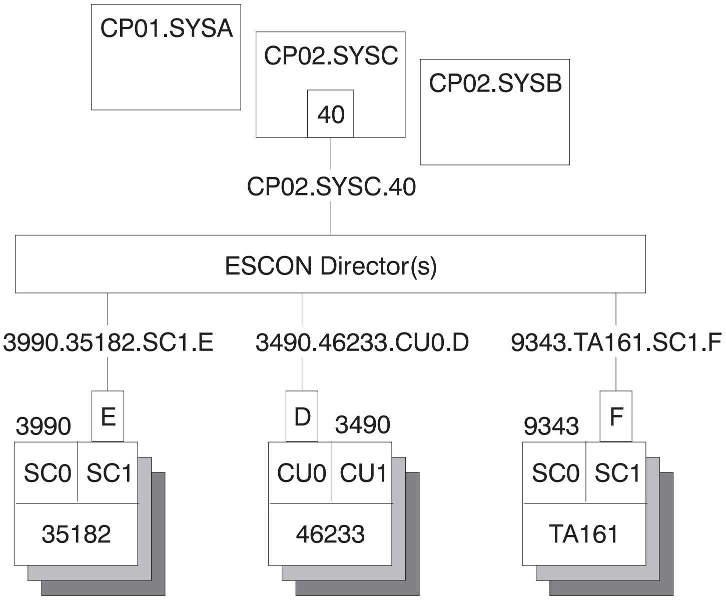


*Figure 14. Examples of Port Names in a Configuration*

## Methods of Naming Ports

You can assign names to ports using the following:

- The WRITE command

  You can use the command:

  ```
  WRITE CP01.SYSB.CHP38 (D3) 100
  ```

  to write the name CP01.SYSB.CHP38 to port D3 on switch 100. This command is available on the operator command line, the ISPF command line, the workstation feature command builder, and the port settings notebook.
- The matrix editor

  You can use the matrix editor to enter a name next to the port number; then send the matrix to the switch. This interface is available on ISPF and the workstation.
- EXECs

  You can create an EXEC with commands like:

  ```
  WRITE CP01.SYSB.CHP38 (D3) 100
  WRITE 3990.35182.SC1.E (F1) 100
  ```

  to send a series of name assignments to a dynamic switch.
- The WRITE switch (WRITESWCH)

  You can create an EXEC to issue the WRITESWCH command, placing the new name in the WRITESWCH data block.

## Using Port Logical Names

Once names are assigned to ports, you can issue a single command to change the connectivity of one or more switches. The command:

```
BLOCK 3490.46233.CU1.F 100
```

blocks the port named 3490.46233.CU1.F on switch 100. The command:

```
BLOCK 3490.46233.CU1.F *
```

blocks the port named 3490.46233.CU1.F on any switch that contains that name. The command:

```
PROHIBIT CP02.SYSC.CHP42 3490.46233.CU1.F *
```

looks for any switch that has both names, CP02.SYSC.CHP42 and 3490.46233.CU1.F. If both names exist on any switch, those two ports are prohibited from each other.

The use of these commands is limited to one change per switch.

## Using Generic Logical Names

SA z/OS I/O operations provides the ability to use an asterisk as a wild card character in commands that use port names. This allows you to make more than one change on each switch.

You can use an asterisk as a name in the DISPLAY NAME, BLOCK, UNBLOCK, ALLOW, and PROHIBIT connectivity commands. For example, if you issue:

```
PROHIBIT CP02* 3490.46233* *
```

all switches are searched for ports with names beginning with CP02 (for example, CP02.SYSA.CHP34 and CP02.SYSB.CHP70) and ports with names beginning with 3490.46233 (for example, 3490.46233.CU1.B and 3490.46233.CU0.D). If found, those ports are prohibited from each other.

By using a single command, you can remove connectivity from a entire system to a control unit. However, for this to work properly:

- The names must be consistent across all switches.
- You must issue the connectivity commands from an I/O operations system that has access to all switches.

Any names that are not an exact match cause no errors. Any switches that are not affected because they were not accessed cause no errors. You only receive notification if:

- No name match is found on any one switch (warning return code).
- No name match is found on any switch (failure return code).

## Command Usage Examples with Generic Logical Names

The following are some examples of how you can issue I/O operations commands using generic logical names:

- Use DISPLAY NAME to show information about the ports specified:

```
DISPLAY NAME CP02.SYSC* *
SWCH                 STATUS  I/O
PORT NAME                  DEVN    LSN   PORT  H B C  P DEF
CP02.SYSC.CHP22            0400    02     C6   O B      CH
CP02.SYSC.CHP39            0100    00     EC          P CHCU
CP02.SYSC.CHP35            0100    00     C5            CH
CP02.SYSC.CHPE0            0200    01     E0            CH
```

- Use DISPLAY NAME to show information about the ports for the 3490 with serial number 46233:

```
DISPLAY NAME 3490.46233* *
SWCH                 STATUS  I/O
PORT NAME                  DEVN    LSN   PORT  H B C  P DEF
```

```
3490.46233.CU0.D           0100    02     C0            CU
3490.46233.CU0.F           0200    01     F6            CU
3490.46233.CU1.A           0300    00     E7          P CU
3490.46233.CU0.C           0400    03     C1            CU
```

- Use BLOCK to remove access to a 3490 with serial number 46233 (four variations):

  ```
  BLOCK  3490.46233.CU0.D  *           (for one port on some switch)
  BLOCK  3490.46233.CU0*   *           (for one CU side)
  BLOCK  3490.46233*       *           (for one CU)
  BLOCK  3490.46233*      100          (for one CU through SW 100)
  ```

  Notice that the first BLOCK command affects only one switch because there should be only one port with the name 3490.46233.CU0.D.
- Use PROHIBIT and then ALLOW to remove access from one host to one 3490 and give access to another host:

  ```
  PROHIBIT  CP02.SYSC*  3490.46233*  *    (affects multiple paths)
  ALLOW     CP01.SYSA*  3490.46233*  *    (affects multiple paths)
  ```
- Use PROHIBIT to remove access from one host to all 9343s to show results:

  ```
  PROHIBIT  CP02.SYSA*  9343*        *
  DISPLAY   NAME        9343*        *
  SWCH                 STATUS  I/O
  PORT NAME                  DEVN    LSN   PORT  H B C   P DEF
  9343.TA161.SC0.A           0100    02     E0          P CU
  9343.TA161.SC0.B           0200    01     E1          P CU
  9343.TA161.SC1.A           0300    00     E2          P CU
  9343.TA161.SC0.C           0400    03     E1          P CU
  ```

In summary, you can use generic logical names to control system connectivity without being concerned about individual ports and switches.

# Part 2. Installation

This part provides instructions for:

- Chapter 8, “Installing SA  z/OS on Host Systems,” on page 71
- Chapter 9, “Installing SA  z/OS on Workstations,” on page 157

# Chapter 8. Installing SA z/OS on Host Systems

This chapter describes the tasks required to install SA z/OS components on the SA z/OS host systems. This chapter includes information on installing SA z/OS on both focal point and target systems. The target system installation does not require some of the steps used for the focal point installation. Any installation step that does not apply to the target systems is indicated. Many of the installation steps have corresponding planning activities and explanations in chapters 2 through 6 of this book. Chapter 9 describes installation on workstations.

In this chapter, the single installation steps are marked as either being required for all or certain SA z/OS components or as being ***optional***. ***Optional*** denotes steps that may or may not need to be performed based on your environment, your system management procedures, and your use of the SA z/OS product. For each of these steps you need to decide whether it is required for your installation.

Each optional step explains why it is optional and describes the circumstances when you will need to perform it.

**Notes:**

1. The meaning of the term *target system* as used by SMP/E needs to be distinguished from the way the term is used in SA z/OS. As used in SMP/E and when describing the installation of z/OS products and services, a target system is the system on which a product such as SA z/OS is installed. It is the collection of program libraries that are updated during SMP/E APPLY and RESTORE processing. In this publication this meaning of target system is referred to as an “SMP/E target system”. The usual SA z/OS meaning of a “target system” is a computer system attached to a focal point system for purposes of monitoring and control.
2. In this book, data set names are shown with the high level qualifier ING. You can have a different high level qualifier for your data sets.
3. If ESCON Manager is already installed, consider that SA z/OS ***cannot*** run together with ESCON Manager on the same system. Running a mixed environment will end up with unpredictable results for example, storage overlay ABEND0C4 or ABEND0C1. See also “Step 4D: Update LPALST*xx*” on page 82 and “Step 4E: Update LNKLST*xx*” on page 83.

## Overview of Installation Tasks

The major tasks required for installing SA z/OS on a focal point are listed in Table 8.

*Table 8. Installation Tasks for SA z/OS Host Systems.* ✓=Required, *=Optional

| Task | SysOps | ProcOps | I/O Ops |
|---|---|---|---|
| “Step 1: SMP/E Installation” on page 74 | ✓ | ✓ | ✓ |
| “Step 2: Allocate System-Unique Data Sets” on page 76 | ✓ | ✓ | ✓ |
| “Step 3: Allocate Data Sets for the Customization Dialog” on page 80 | ✓ | ✓ | ✓ |
| “Step 4: Customize SYS1.PARMLIB Members” on page 81 | ✓ | ✓ | ✓ |
| “Step 5: Setting up WebSphere MQ” on page 85 | ✓ | | |
| “Step 6: Customize SYS1.PROCLIB Members” on page 87 | ✓ | ✓ | ✓ |
| “Step 7: Customize NetView” on page 89 | ✓ | ✓ | |
| “Step 8: Preparing the Hardware” on page 97 | ✓ | ✓ | |
| “Step 9: Preparing the VM PSM” on page 106 | * | | |
| “Step 10: Customizing the Automation Manager” on page 109 | ✓ | | |
| “Step 11: Customizing the Component Trace” on page 111 | ✓ | ✓ | |
| “Step 12: Customizing the System Logger” on page 111 | * | | |
| “Step 13: Install ISPF Dialog Panels” on page 113 | ✓ | ✓ | ✓ |
| “Step 14: Verify the Number of available REXX Environments” on page 119 | ✓ | ✓ | |
| “Step 15: Customization of NetView for TEC Notification by SA z/OS” on page 119 | * | * | |

Table 8. Installation Tasks for SA z/OS Host Systems (continued). ✓=Required, *=Optional

| Task | SysOps | ProcOps | I/O Ops |
|---|---|---|---|
| "Step 16: Compile SA z/OS REXX Procedures" on page 122 | * | * | |
| "Step 17: Defining Automation Policy" on page 123 | ✓ | ✓ | |
| "Step 18: Define Host-to-Host Communications" on page 124 | ✓ | ✓ | ✓ |
| "Step 19: Enabling SA z/OS to Restart Automatic Restart Manager Enabled Subsystems" on page 129 | ✓ | | |
| "Step 20: Define Security" on page 130 | ✓ | ✓ | |
| "Step 21: Customize the Status Display Facility (SDF)" on page 131 | * | | |
| "Step 22: Check for Required IPL" on page 131 | ✓ | ✓ | ✓ |
| "Step 23: Automate System Operations Startup" on page 131 | ✓ | | |
| "Step 24: Verify Automatic System Operations Startup" on page 133 | * | | |
| "Step 25: Install an SA z/OS Satellite" on page 134 | * | | |
| "Step 26: Installing and Customizing the NMC Focal Point" on page 136 | * | | |
| "Step 27: Copy and Update Sample Exits" on page 144 | * | * | * |
| "Step 28: Install CICS Automation in CICS" on page 145 | * | | |
| "Step 29: Install IMS automation in IMS" on page 147 | * | | |
| "Step 30: Install TWS Automation in TWS" on page 148 | * | | |
| "Step 31: Install USS Automation" on page 151 | * | | |
| "Step 32: Customizing GDPS" on page 153 | * | | |
| "Step 33: Customizing I/O Operations" on page 155 | | | ✓ |
| "Step 34: Installing Tivoli Enterprise Portal Support" on page 156 | * | | |

## Step 1: SMP/E Installation

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | ✓ |

Perform the SMP/E installation as described in the *Program Directory* document shipped with this product. This documentation contains the required information on how to build the SMP/E environment.

**Note:** In the steps that follow, sample jobs are all members of the SINGSAMP data set, the SA z/OS sample library.

Table 9 shows a list of target data sets as provided by the SMP/E installation process to be used for production on your system.

*Table 9. Target Data Sets*

| Data Set Name | Description |
|---|---|
| ING.SINGIMSG | ISPF messages 1 |
| ING.SINGINST | SMP/E jobs to install the product alternatively to using SMP/E dialogs 2 |
| ING.SINGIPDB | Policy database samples 1 |
| ING.SINGIPNL | ISPF panels 1 |
| ING.SINGIREX | ISPF REXX execs 1 |
| ING.SINGISKL | ISPF skeletons 1 |
| ING.SINGITBL | ISPF tables 1 |
| ING.SINGJMSG | Kanji NetView messages 5 |
| ING.SINGJPNL | Kanji NetView panels 5 |
| ING.SINGMOD1 | Different SA z/OS modules 3 |
| ING.SINGMOD2 | Different SA z/OS modules in LINKLST 3 |
| ING.SINGMOD3 | Different SA z/OS modules in LPALIB 3 |
| ING.SINGNMSG | NetView messages 3 |
| ING.SINGNPNL | NetView panels 3 |
| ING.SINGNPRF | NetView profiles 3 |
| ING.SINGNPRM | NetView DSIPARM samples 3 |
| ING.SINGNREX | NetView REXX execs 3 |
| ING.SINGSRC | SA z/OS source 3 |
| ING.SINGPWS1 | NMC exploitation code 4 |
| ING.SINGJPWS | Japanese NMC exploitation code 5 |
| ING.SINGSAMP | General samples 3 |
| ING.SINGMSGV | For VM second level systems support 6 |
| ING.SINGOBJV | For VM second level systems support 6 |
| ING.SINGREXV | For VM second level systems support 6 |
| *&shilev*.TKANCUS | Installation CLISTs for Tivoli Enterprise Portal (TEP) support 7 |
| *&shilev*.TKANMODL | Load modules for TEP support 7 |
| *&shilev*.TKANDATV | Data files for TEP support 7 |
| *&shilev*.TKANPAR | Parameter files for TEP support 7 |

Table 10 shows a list of the HFS directories that are provided by the SMP/E installation process.

*Table 10. HFS Paths*

| HFS Path | Description |
|---|---|
| /usr/lpp/ing/adapter | Shell script 8 |
| /usr/lpp/ing/adapter/lib | Executable 8 |

*Table 10. HFS Paths (continued)*

| HFS Path | Description |
|---|---|
| /usr/lpp/ing/adapter/config | Configuration file 8 |
| /usr/lpp/ing/adapter/data | Customer data/empty at installation 8 |
| /usr/lpp/ing/adapter/ssl | Customer data/empty at installation 8 |
| /usr/lpp/ing/ussauto | Customer data/empty at installation 8 |
| /usr/lpp/ing/ussauto/lib | USS automation executable 8 |
| /usr/lpp/ing/doc | Documentation |
| /usr/lpp/ing/doc/policies | Best practice policy diagrams |

The following list helps you to grant RACF access to the appropriate users of the data sets:

**1** Data sets of this category are related to ISPF and need to be accessed by everyone that uses the customization dialog.

**2** Data sets of this category need to be accessed by the system programmer running SMP/E.

**3** Data sets of this category need to be used by the NetView and automation team responsible for setting up and customizing system automation and I/O operations.

**4** Data sets of this category need to be accessed by anyone who will be installing the NMC component.

**5** Data sets of this category are only required if you install Kanji support.

**6** Data sets of this category are defined in VM setup.

**7** These data sets are required for Tivoli Enterprise Portal support, where *&shilev* is the high-level qualifier of the SMP/E target libraries used. See also *IBM Tivoli System Automation for z/OS Monitoring Agent Configuration and User's Guide*.

**8** Files in these directories are used for USS Automation and the end-to-end automation adapter.

## Step 2: Allocate System-Unique Data Sets

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | | ✓ |

Certain data sets are required several times across the focal point and target systems. This section tells you which are required on which systems or sysplexes. To allocate these data sets, sample jobs are provided in the following members of the SINGSAMP data set:

- INGALLC0
- INGALLC1
- INGALLC2
- INGALLC3
- INGALLC4
- INGALLC5

- INGALLC6

**Prerequisite for running the jobs:**

Before you run these jobs, you need to edit them to make them runnable in your specific environment. To do so, first copy them into your private user library and then follow the instructions that are given in the comments in the jobs.

Note that the values that you fill in (such as the system name) may be different for each system where you run the jobs.

## Step 2A: Data Sets for NetView

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | | |

The data sets in Table 11 are required once per automation agent and cannot be shared between automation agents. They need to be referred to in the startup procedure for each automation agent's NetView in "Step 6: Customize SYS1.PROCLIB Members" on page 87.

*Table 11. Data Sets for Each Individual Automation Agent*

| Purpose | Sample job to allocate the data set | Organization | DD name in the NetView startup procedure |
|---|---|---|---|
| User-modified NetView system definitions. | INGALLC0 | Sequential | DSIPARM |
| Stores the NetView reports, listings, files, and output from the security migration tool as well as the reports from the style sheet report generator. | INGALLC0 | Sequential | DSILIST |
| Contains the members to be used when testing the automation table. | INGALLC0 | Sequential | DSIASRC |
| Stores the output report produced from running tests of the automation table. | INGALLC0 | Sequential | DSIARPT |
| Contains VTAM source definitions for the sample network. | INGALLC0 | Sequential | DSIVTAM |
| NetView log data sets | INGALLC0 | VSAM | DSILOGP, DSILOGS |
| NetView trace data set | INGALLC0 | VSAM | DSITRCP, DSITRCS |
| NetView save/restore data set | INGALLC0 | VSAM | DSISVRT |

## Step 2B: Data Sets for I/O Operations

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| | | ✓ |

### Step 2: Allocate System-Unique Data Sets

The data set in Table 12 is required once on each system where you want to have I/O operations available. It cannot be shared between systems. It needs to be referred to in the I/O operations startup procedure in “Step 6: Customize SYS1.PROCLIB Members” on page 87.

*Table 12. Data Sets for I/O Operations*

| Purpose | Sample job to allocate the data set | Organization | DD name in the I/O operations startup procedure |
|---|---|---|---|
| HCD trace file | INGALLC1 | Sequential | HCDTRACE |

## Step 2C: Data Sets for Automation Agents

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | | |

The data sets in Table 13 are required once per automation agent and cannot be shared between automation agents. They need to be referred to in the startup procedure for each automation agent’s NetView in “Step 6: Customize SYS1.PROCLIB Members” on page 87.

*Table 13. Data Sets for Each Individual Automation Agent*

| Purpose | Sample job to allocate the data set | Organization | DD name in the NetView startup procedure |
|---|---|---|---|
| Automation status file | INGALLC2 | VSAM | AOFSTAT |
| Dump file for diagnostic information | INGALLC2 | Sequential | INGDUMP |

The data set in Table 14 is required once per sysplex and cannot be shared across sysplex boundaries. It needs to be referred to in the startup procedure for each automation agent’s NetView in “Step 6: Customize SYS1.PROCLIB Members” on page 87.

*Table 14. Data Set for Each Sysplex*

| Purpose | Sample job to allocate the data set | Organization | DD name in the NetView startup procedure |
|---|---|---|---|
| IPL data collection | INGALLC4 | VSAM | HSAIPL |

## Step 2D: Data Sets for Automation Managers (Primary Automation Manager and Backups)

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | | |

The data sets in Table 15 on page 79 are required once per sysplex or standalone system. In the same sysplex or standalone system, they should be shared by the primary automation manager and its backups, but they cannot be shared across sysplex or standalone-system boundaries. Except for the takeover file, they need to be referred to in the automation manager startup procedure in "Step 6: Customize SYS1.PROCLIB Members" on page 87.

Each subplex requires one separate set of the following:
- The schedule override file
- The configuration information data set
- The automation manager takeover file

*Table 15. Data Sets for All Automation Managers in a Sysplex or Standalone System*

| **Purpose** | **Sample job to allocate the data set** | **Organization** | **DD name in the automation manager startup procedure** |
|---|---|---|---|
| Schedule override file | INGALLC3 | VSAM | HSAOVR |
| Configuration information data set | INGALLC3 | Sequential | HSACFGIN |
| PARMLIB | INGALLC3 | Partitioned | HSAPLIB |
| Takeover file | INGALLC3 | VSAM | — |
| **Note:** Use the following formula to work out the required size of the takeover file: `4000 records + *n* records of 4K`, where *n* is the maximum numbers of resources. | | | |

The data sets in Table 16 must be allocated once for each automation manager. They cannot be shared between an automation manager and its backups on the same system. Therefore, when you edit the sample job that is to allocate the data sets for a particular sysplex or standalone system, make sure that you include a fresh job step for each automation manager that you plan to have on that particular sysplex or standalone system. For more details, see the comments in the INGALLC3 sample.

**Note:** You can safely use the same DD names in each job step because DD names are not shared across job step boundaries.

These files also need to be referred to in the automation manager startup procedure in "Step 6: Customize SYS1.PROCLIB Members" on page 87.

*Table 16. Data Sets for Each Individual Automation Manager*

| **Purpose** | **Sample job to allocate the data set** | **Organization** | **DD name in the automation manager startup procedure** |
|---|---|---|---|
| Internal trace files (optional) | INGALLC5 | Sequential | TRACET0 |
| | INGALLC5 | Sequential | TRACET1 |
| ALLOCOUT data set | INGALLC5 | Sequential | SYSOUT |
| ALLOCPRT data set | INGALLC5 | Sequential | SYSPRINT |
| DUMP data set for LE environment | INGALLC5 | Sequential | CEEDUMP |

The generation data groups (GDGs) in Table 17 on page 80 must be created once for each automation manager. They cannot be shared between an automation manager and its backups on the same system. Therefore, when you edit the sample job that is to create the GDGs for a particular sysplex or standalone system, make sure that you include a new set of GDG definitions for each automation manager that you plan to have on that particular sysplex or standalone system. For more details, see the comments in the INGALLC6 sample.

### Step 2: Allocate System-Unique Data Sets

These files also need to be referred to in the automation manager startup procedure in “Step 6: Customize SYS1.PROCLIB Members” on page 87.

*Table 17. Generation Data Groups for Each Individual Automation Manager*

| Purpose | Sample job to create the GDG | Organization | DD name in the automation manager startup procedure |
|---|---|---|---|
| Internal trace files | INGALLC6 | Sequential | TRACET0 |
| | INGALLC6 | Sequential | TRACET1 |
| ALLOCOUT data set | INGALLC6 | Sequential | SYSOUT |
| ALLOCPRT data set | INGALLC6 | Sequential | SYSPRINT |
| DUMP data set for LE environment | INGALLC6 | Sequential | CEEDUMP |

## Step 3: Allocate Data Sets for the Customization Dialog

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | ✓ |

Use the sample job INGEDLGA in SINGSAMP to allocate data sets that are required for I/O operations and the customization dialog. These data sets are normally allocated only on the focal point system where you use the customization dialog. These data sets include:

- **For system operations:**
  - The ISPF table library data set that contains the values you enter in the customization dialog
  - The system operations control file: this is the output data set for the customization dialog when building the system operations control files (automation control file and automation manager configuration file)

| Data Set Name | Purpose |
|---|---|
| ING.CUSTOM.AOFTABL | ISPF customization table for customization dialog |
| ING.CUSTOM.SOCNTL | System operations control file |

- **For processor operations:**
  - The ISPF table library data set that contains the values you enter in the customization dialog
  - The processor operations control file, generated using the customization dialog, which provides information about your processor operations configuration
  - The processor operations control file log, which receives messages that result from generating the processor operations control file

| Data Set Name | Purpose |
|---|---|
| ING.CUSTOM.AOFTABL | ISPF customization table for customization dialog |
| ING.CUSTOM.POCNTL | Processor operations control file |
| ING.CUSTOM.POLOG | Processor operations control file log |

- **For I/O operations:**

- The I/O operations configuration file. Because you use the customization dialog to collect information and build control files, you normally need them only at the focal point. The I/O operations dialogs, however, are used to input commands and get responses from the I/O operations part of SA z/OS. Because they do not support multisystem commands for I/O operations functions, you must install them on each system, focal point or target, where you want to use them.

| Data Set Name | Purpose |
|---|---|
| ING.CUSTOM.IHVCONF | I/O operations configuration file |

**Note:** Make a note of these data set names. They are used in "Step 13: Install ISPF Dialog Panels" on page 113. If you rename the data sets, you need to adapt the corresponding names in that step.

## Step 4: Customize SYS1.PARMLIB Members

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | ✓ |

The *xx* suffix on each SYS1.PARMLIB data set member can be any two characters chosen to match your IEASYS naming scheme. The SA z/OS samples delivered in SINGSAMP use a suffix of *SO*. See *z/OS MVS Initialization and Tuning Reference* for information about IEASYS.

The following sections describe the SYS1.PARMLIB data set members that need to be changed and provide information on how to achieve this.

### Step 4A: Update PROG*xx*

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | ✓ |

With DFSMS/MVS®, you can define authorized libraries in a PROG*xx* member for dynamic authorized program facility (APF). You can activate a PROG*xx* list using the SET PROG=*xx* command without IPLing the system. Alternatively, you can define authorized libraries to the APF in an IEAAPF*xx* member. For a complete description of dynamic APF and PROG*xx*, see *z/OS MVS Initialization and Tuning Reference*.

Update PROG*xx* to include:

- ING.SINGMOD1, ING.SINGMOD2, ING.SINGMOD3
- SYS1.SCBDHENU (for I/O operations)

  **Note:** Do not include SYS1.NUCLEUS.
- If you chose to set AOF_SET_AVM_RESTART_EXIT to 0 in the NetView style sheet and use ARM to restart resources, add the following entry to your PROG*xx* member:

```
EXIT ADD
     EXITNAME(IXC_ELEM_RESTART)
     MODNAME(AOFPERRE)
```

## Step 4B: Update SCHED*xx*

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ |  | ✓ |

Compare the content of the SCHED*xx* member with the INGESCH member that resides in the SINGSAMP sample library. Edit the SCHED*xx* member so that it includes all the statements in the INGESCH member.

This enables the NetView subsystem interface address space, the NetView application address space (for the automation agent), the I/O operations address space and the automation manager to run without being swapped out of memory.

I/O operations exploits the MVS component trace and stores intermediate trace records in a data space. Because some trace entries are recorded outside the I/O operations address space, the data space must be common to all users. However, a common data space requires the owning address space to be non-swappable.

## Step 4C: Update MPFLST*xx*

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ |  |

It is recommended that you update the MPFLST*xx* member *after* having installed the ISPF Customization Dialog (see "Step 17: Defining Automation Policy" on page 123). Using the customization dialog you can define your Automation Policy and create a list of messages involved in automation. The customization dialog also allows you to define header and trailer lines for the message list, thus building a complete MPFLST*xx* member called MPFLSTSA.

Alternatively, update the contents of the MPFLST*xx* member with the INGEMPF member that resides in the SINGSAMP sample library. Edit the MPFLST*xx* member so that it includes all the statements in the INGEMPF member. Review the MPFLST*xx* member to ensure that it is appropriate for your system, and resolve any conflicts.

This adds the SA z/OS message automation and console display suppression specifications to the MPFLST*xx* member.

Make sure that messages for any products that you want to automate (that is, CICS Automation, TWS Automation, etc.) are forwarded to automation.

## Step 4D: Update LPALST*xx*

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | ✓ |

Edit the LPALST*xx* member to add ING.SINGMOD3 to the SA z/OS load library. There is no other choice for this library, it must be in the LPALST concatenation.

**You can avoid an IPL:**

Because ING.SINGMOD3 contains only a few modules, you can also code a PROG*xx* member that enables a dynamic addition of those modules to the LPALST. If you do this, no IPL is required.

## Step 4E: Update LNKLST*xx*

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | ✓ |

To run SA z/OS, you must ensure that program libraries can be found at startup time.

Add SINGMOD1 (recommended) and SINGMOD2 (mandatory) to the LNKLST concatenation. There is no other choice for these libraries: they **must** be in the LNKLST concatenation.

For the other libraries, either add them to the LNKLST concatenation or add them on STEPLIB DDs in the JCL in SYS1.PROCLIB that is used to start the products.

Adding libraries on STEPLIB DDs will involve performance degradation compared to adding them to the LNKLST concatenation and should therefore be avoided.

z/OS link list data sets no longer have to be cataloged in the master catalog. It is possible to specify a volume in the link list entry for data sets that are cataloged in user catalogs.

## Step 4F: Update IEFSSN*xx*

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | |

Ensure that IEFSSN*xx* contains all the statements in the INGESSN sample member. If this has already been accomplished during the NetView installation there are no further updates required to this member.

Compare the contents of the IEFSSN*xx* member with the INGESSN member, which resides in the SA z/OS sample library. Edit the IEFSSN*xx* member so that it includes the subsystem records from the INGESSN member.

This defines:

- Four-character prefix used in the NetView startup procedure member names. The four-character prefix that you specify must match the four-character prefix of the NetView startup procedure member names. For example, if you specify INGE, the names of the NetView startup procedure members must be INGE*xxxx*, where *xxxx* are any four characters you choose. If you change this four-character prefix, you can dynamically add this entry using the z/OS command SETSSI. Otherwise you must perform an IPL of z/OS to effect the change.

- JES startup specifying JES2 or JES3 with the NOSTART option. This prevents JES from starting before SA z/OS during the IPL process. If you plan to start JES before NetView, remove the NOSTART option from the following statement:

  ```
  JESx,,,PRIMARY,NOSTART
  ```

  You can also use the IEFSSN-syntax:

  ```
  JESx,PRIMARY(YES),START(NO)
  ```

  The positional syntax (PRIMARY,NOSTART) is still supported. For the correct syntax of your environment check the *z/OS MVS Initialization and Tuning Reference*.

The first active NetView SSI is used for program-to-program interface communication. When a NetView SSI is active and in use by the program-to-program interface (PPI), and another NetView SSI becomes active that is coded higher in the SSN table, then the PPI will switch and use that NetView SSI. If product automation has already signed on to the PPI before the switch occurs, product automation program-to-program communications will be disrupted.

To ensure that disruptions do not occur, do one of the following:

- Make sure that the SA z/OS SSI entry is the first SSI in the SSN table and the SSI starts during the IPL.
- Use an option available with NetView to specify “NOPPI” on all NetView SSIs except the SSI that product automation uses. This “NOPPI” option is specified as a startup parameter on the SSI JCL.
- If you do not code the SSI that product automation uses in the highest position in the SSN table and you do not use the “NOPPI” option, the SSI that is first in the SSN table must be up before product automation initialization and must remain uninterrupted until final termination of product automation.

Check the subsystem name table in MVS SYS1.PARMLIB member IEFSSN*xx* to verify that the NetView SSI used by product automation is first in the list (ahead of all other NetView subsystem names).

## Step 4G: Update JES3IN*xx*

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | | |

If you are using JES3, compare the contents of the JES3IN*xx* member with the INGEJES3 member which resides in the SINGSAMP sample library. You may want to review these members first to see whether there are entries in the INGEJES3 member that are already in the JES3IN*xx* member. After merging the INGEJES3 member, be sure there are no duplicate entries in the JES3IN*xx* member.

This includes the DUMP options and adds the JES3 parameters.

## Step 4H: Update SMFPRM*xx*

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| * | | |

If you plan to use SMF records for availability reporting you must update the SMFPRM*xx* member in the SYS1.PARMLIB library by adding type 114 to the SYS(TYPE statement :

```
SYS(TYPE(30,...,114)
```

## Step 5: Setting up WebSphere MQ

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | | |

If you want to use WebSphere MQ for communication between the automation manager and the automation agents and provide a continuous high-reliable environment for the automation manager, you must set up an WebSphere MQ manager. The basic steps to do this are described in *WebSphere MQ for z/OS System Management Guide*. The outline for setting up WebSphere MQ for exploitation by SA z/OS is described in the following substeps.

**Note:** This step is not necessary when you have decided to use XCF for communication between the automation manager and the automation agents.

### Step 5A: Customizing a WebSphere MQ Manager for SA z/OS

You need to carry out this substep on every system where either the automation manager or an automation agent, or both, is installed.

Refer to *WebSphere MQ for z/OS System Management Guide* for the correct WebSphere MQ installation and setup processing. It is recommended that there is a single WebSphere MQ manager instance for SA z/OS. The way SA z/OS exploits the WebSphere MQ infrastructure does not immediately require a dedicated DB2 for WebSphere MQ's shared data repository.

The following list describes which parameters and WebSphere MQ options need special consideration for SA z/OS:

- SA z/OS is using the TSO/Batch adapter.
- SA z/OS does not require any distributed queuing capabilities.
- Archiving can be switched off. See Macro CSQ6LOGP.
- The maximum number of connections SA z/OS is using is in the range of 20 which is the current default. However if the number of SA z/OS query threads is increased drastically you may also require additional connections. See Option IDBACK in Macro CSQ6SYSP.
- The maximum number of messages processed per WebSphere MQ transaction is normally set to 10000 via CSQINP1. See WebSphere MQ DISPLAY MAXSMSGS. This should be sufficient in all cases. However when it turns out that the number of messages in the State Queue (see INGAMS commands) reaches the area of 4000, this value should be set to approximately. 2.5 * the number of expected state queue messages. See WebSphere MQ DEFINE MAXSMSGS.
- For the calculation of the number of log records see the *WebSphere MQ for z/OS System Management Guide*. You may consider that the largest processing load which is to be logged is the takeover case, where about 5000 MQPUTs and 5000 MQGETs with an average of 4 K messages are produced in a time frame of one minute. As a rule of thumb, you can take the MAXSMSGS value, divide it by two to get the number of maximal MQGETs. The same number can be taken for

the maximum number of MQPUTS. The largest transaction producing so many GETs and PUTs should be in a range of a minute and processes 4 K messages.

### ARM Considerations for WebSphere MQ Manager

If you choose z/OS Automatic Restart Manager for doing the restart, the WebSphere MQ manager instance must be set up to allow element restarts only. A cross system restart is not required.

## Step 5B: Definition of CF Structures for a Sysplex Environment

In a full sysplex environment, you need to build CF list structures needed for WebSphere MQ shared queues. One CF list structure can have more than one WebSphere MQ queue, however a queue cannot span CF structure boundaries. It is recommended that you use two CF structures for the SA z/OS queues. The Work Item Queue and the Agent Queue can easily share a CF structure. The CF storage size consumption for the Automation State Queue could be very dynamic, because the automation manager can generate a huge amount of uncommitted WebSphere MQ messages also using CF storage.

Refer to the *WebSphere MQ for z/OS System Management Guide* for information on how to calculate the size of the CF List Structures. The number of messages per queue and the message size can be taken from the provided samples (INGALLMS, INGALLML). Keep in mind that you have to double the number of messages for the state queue because of uncommitted updates which also occupy CF storage.

## Step 5C: Definition of WebSphere MQ Queues

Two sample jobs are delivered defining the queues either as

- Local Queues for a single system environment with Sample INGALLML
- Shared Queues for a full Sysplex Environment with Sample INGALLMS

Important operands to consider:

**name of the queue**

The names of the queues are fixed and follow the following pattern:

- 'HSA' the SA z/OS automation manager component prefix
- The XCF group ID (8 Characters) to allow more than one Automation domain per Sysplex

  This is the only modifiable part. Refer to the samples.
- A predefined character string as suffix:
  - WORKITEM.QUEUE for the Work Item Queue
  - STATE.QUEUE for the Automation State Queue
  - AGENT.QUEUE for the Automation Agent Queue

**CFSTRUCT**

Name of the CF List structure used by this queue. This is only valid for shared queues.

**MAXMSGL**

maximum length in bytes per message (excluding messages descriptor). This length plus the message descriptor should not be larger than 4K. This value is already provided in the samples and should not be changed

**MAXDEPTH**

maximum possible number of messages. The samples provides values which should fit your environment. INGAMS can be used to monitor whether there is a danger that a queue becomes full. In this situation the size should be changed accordingly.

### Display WebSphere MQ Statistics

Operators may be interested in some WebSphere MQ queue statistics. The INGAMS command with the automation manager detail option provides the data.

It is also possible to use the ISPF based WebSphere MQ standard operations and control panel to operate with the SA z/OS queues.

## Step 5D: RACF Considerations for WebSphere MQ

If you are using RACF to protect WebSphere MQ resources on your system the following RACF profiles must be defined:

```
CLASS(MQCONN) profile mqsubsysid.BATCH
CLASS(MQCONN) profile qsgname.BATCH
CLASS(MQQUEUE) profile mqsubsysid.HSA.**
```

**Notes:**

1. mqsubsysid is the 4 character subsystem name of the local WebSphere MQ queue manager.
2. qsgname is the name of the queue-sharing group to which the queue manager belongs (refer to the QSGDATA statement in the WebSphere MQ documentation for details on qsgname.
3. The SA z/OS automation manager and automation agents must be granted READ access to the resource profiles of MQCONN.
4. The SA z/OS automation manager and automation agents must be granted ALTER access to the resource profiles of MQQUEUE.

# Step 6: Customize SYS1.PROCLIB Members

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | ✓ |

Some changes need to be made to startup procedure members in the SYS1.PROCLIB data set. It is recommended that you either back up the startup procedure members that you are going to change or that you create new members.

## Step 6A: NetView Startup Procedures

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | |

- **NetView Subsystem Interface Startup Procedure**

  NetView provides a sample subsystem interface startup procedure in member CNMSJ010. Copy this member from your NetView library and adapt it to your needs. Rename it to agree with the four-character prefix defined in the IEFSSN*xx* member that is described in “Step 4F: Update IEFSSN*xx*” on page 83.
- **NetView Application Startup Procedure**

  You can use the sample provided in the INGENVSA member of the SINGSAMP data set. Copy it to a member of each system’s SYS1.PROCLIB data set (for the focal point system as well as for the target systems).

Customize each copy to your needs. In particular, do the following:

1. Make sure that the AOFSTAT, INGDUMP and HSAIPL concatenations include the data sets that you allocated in "Step 2: Allocate System-Unique Data Sets" on page 76.
2. Rename the NetView application startup procedure member to agree with the four-character prefix defined in the IEFSSN*xx* member, which is described in "Step 4F: Update IEFSSN*xx*" on page 83. For example, if the name of the NetView application startup procedure is INGE*xx*, INGE must be specified in the IEFSSN*xx* member as the character prefix.

If you do not make ING01 your domain name, make a note of what your NetView domain name is. This information is needed in "Step 6C: I/O Operations Startup Procedure" on page 89 and for system operations. See also *IBM Tivoli System Automation for z/OS Defining Automation Policy* for more information on enterprise definitions.

See *Tivoli NetView for z/OS Installation: Configuring Additional Components* for further details on how to modify the NetView startup procedure.

## Step 6B: Startup Procedures Required for System Operations Only

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | | |

- **Automation Manager Startup Procedure**

  You can use the sample provided in the INGEAMSA member of the SINGSAMP data set. Copy it to a member of the SYS1.PROCLIB data set of the focal point system.

  Customize that copy to your needs. In particular, make sure that the DD concatenations mentioned in "Step 2: Allocate System-Unique Data Sets" on page 76 include the data sets that you allocated there. In addition, consider customizing the following points:

  – If you prefer not to place the automation manager PARMLIB member in the SYS1.PARMLIB concatenation, include a HSAPLIB DD statement in the automation manager startup procedure (see also "Step 10: Customizing the Automation Manager" on page 109):

  ```
  HSAPLIB DD DSN=ING.PARMLIB, DISP=SHR
  ```

  In place of `ING.PARMLIB`, use the PARMLIB data set that you allocated in "Step 2: Allocate System-Unique Data Sets" on page 76.

  – A separate NON-APF authorized task library is required in addition to the authorized STEPLIB.

    - The NON-APF authorized task library is used by the LE task. It must concatenate the NON-APF authorized SA z/OS product library with the LE runtime library and the C/C++ library.
    - The STEPLIB concatenation specifies the APF-authorized SA z/OS product library.

- **Other System Operations Startup Procedures**

  Copy the following members from the SINGSAMP data set to members of the SYS1.PROCLIB of the focal point system:

  – INGPIXCU

  – INGPHOM

- INGPIPLC
- HSAPIPLC

Follow the customization instructions that are contained in the HSAPIPLC member.

**Note:** These procedures make use of certain data sets and must have the appropriate authorizations. For details refer to “Granting NetView and the STC-User Access to Data Sets” on page 171.

- *Optional:* **Startup Procedure for the External Writer of the Component Trace**

  Copy member HSACTWR from SINGSAMP. At least the SYSNAME parameter must be specified before the procedure is stored in a library of the PROCLIB concatenation.

### Step 6C: I/O Operations Startup Procedure

| SysOps | ProcOps | I/O Ops |
|---|---|---|
|  |  | ✓ |

You can use the sample provided in the INGEIO member of the SINGSAMP data set. Copy it to a member of each system’s SYS1.PROCLIB data set (for the focal point system as well as for the target systems).

Customize those copies to your needs. In particular, do the following:

- Specify the NetView domain name that you made a note of in step “Step 6A: NetView Startup Procedures” on page 87.
- Make sure that the HCDTRACE concatenation in the procedure includes the data set that you allocated for I/O operations in “Step 2: Allocate System-Unique Data Sets” on page 76.

Due to the fact that z/OS 1.4 HCD has changed the default of the profile option IODF_DATA_SPACE from NO to YES, it is no longer necessary to define the HCD profile data set for I/O operations. However, if you need to specify options for HCD tracing, refer to “Defining an HCD profile” in the *z/OS HCD User’s Guide* for how to create that data set.

## Step 7: Customize NetView

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ |  |

This section discusses how to customize several aspects of NetView:

- “Step 7A: Customize NetView Alert Information” on page 90
- “Step 7B: Customize NetView DSIPARM Data Set” on page 90
- “Step 7C: Modifying NetView DSIPARM Definitions for an Automation Network” on page 95
- “Step 7D: Customize NetView for Processor Operations” on page 95
- “Step 7E: Customize the NetView Message Translation Table” on page 96

## Step 7A: Customize NetView Alert Information

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | | |

SA z/OS enterprise monitoring depends upon alert information being passed from remote systems to the focal point. Note that this is only necessary when communication is via NPDA alerts.

The NetView command SRFILTER (or SRF) establishes the conditions governing the recording of data in the hardware monitor database, the generation of messages to the authorized operator, the forwarding of alert data to a NetView focal point, and the coloring of alerts.

To ensure that the alerts required by SA z/OS for enterprise monitoring are not filtered out, the following is recommended:

- On any focal point system:
  - Issue the command: `SRF AREC PASS N *`
- From the remote systems:
  - Issue the command: `SRF AREC PASS N *`
  - Issue the command: `SRF ROUTE CLEAR`

These SRF commands should be included in a startup CLIST or exit because they need to be issued after every NetView startup.

If you do not want to use the `SRF AREC PASS N *` command to allow *all* alerts to pass, you should, as a minimum, allow the NTFY event type (*etype*s) to pass.

The NetView SRFILTER command is documented in *Tivoli NetView for z/OS Command Reference Vol. 1*.

## Step 7B: Customize NetView DSIPARM Data Set

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | |

A sample is provided for this step in the INGSTGEN member of the SINGSAMP library. Copy this to DSIPARM data set, then rename and customize it to match your installation. See INGSTGEN for further details.

Copy any DSIPARM and SINGNPRM member that you need to customize into a data set allocated in DSIPARM before the SMP/E-maintained NetView DSIPARM and SA z/OS target libraries and edit it there.

Then change the following members in the copied NetView DSIPARM data set:

**NetView Style Sheet**

**Tower Statements:** The various SA z/OS components or environments are activated with the following TOWER.SA statements.

**SysOps**

This enables application or more general resource automation.

**ProcOps**
This enables Processor Operations.

**Satellite**
This indicates that the SA z/OS topology manager runs on the Networking NetView for communication with RODM and the NMC.

**GDPS** This enables Geographically Dispersed Parallel Sysplex (GDPS®) to run under SA z/OS. Use this definition regardless of the specific GDPS product that is running (GDPS/PPRC, GDPS/PPRC HM, GDPS/XRC or GDPS/GM).

Additionally the following GDPS subtowers are available to distinguish between the GDPS product running on the system:

| | |
|---|---|
| **PPRC** | For GDPS/PPRC |
| **HM** | For GDPS/PPRC HM |
| **XRC** | For GDPS/XRC |
| **GM** | For GDPS/GM |

Furthermore, code one of the following indicating whether or not this is the production versus K-system:

- PROD for a production system
- KSYS for a K-system

This information is used by SA z/OS to pick up the appropriate definition members that vary for the GDPS controlling system (K system) and the production system. For example, the K system constitutes a subplex of its own and must therefore use a different XCF group name.

**Note:** You must have one of the following installed to specify the GDPS subtower:
- GDPS/PPRC V3.4
- GDPS/PPRC HM V3.4
- GDPS/XRC V3.4
- GDPS/GM V3.4

See the INGSTGEN sample for further details about the SA tower statements.

To enable SA z/OS, make sure that the following TOWER statements are activated in the NetView style sheet (that is, uncomment them):

```
TOWER =  SA
TOWER.SA  = SYSOPS
```

**Kanji Support:** If you plan to use Kanji support make sure that you update the NetView style sheet as follows:

1. `transTbl =DSIKANJI` must be specified.
2. `transMember =CNMTRMSG` must be uncommented.

For more details, refer to the chapter “Installing the National Language Support Feature” in *Tivoli NetView for z/OS, Installation: Configuring Additional Components*.

**Sample Automation Operator AUTO2**: AUTO1 and AUTO2 are sample automation operators that are used in the initialization of SA z/OS and they cannot be used by NetView for NETCONV sessions or resource discovery. To prevent the AUTO2 sample automation operator being used by NetView, do the following in the style sheet:

1. For NETCONV sessions, blank out AUTO2 in the following statement:

   ```
   function.autotask.NetConv = AUTO2
   ```

   The statement should then be:

   ```
   function.autotask.NetConv =
   ```

2. For resource discovery, choose an autotask *other* than AUTO2 in the following statement:

   ```
   function.autotask.autoip = AUTO2
   ```

Refer to the NetView documentation for details about customizing the NetView style sheet.

**AOFMSGSY (optional)**

If you have renamed any automation tasks in AOFOPFxx, you will need to make corresponding changes to the AOFMSGSY member.

Copy and edit the AOFMSGSY member that resides in ING.SINGNPRM and do the following:

1. If you want to define actions for messages that the SA z/OS NetView Automation Table does not trigger any actions for, you can use the symbol %AOFALWAYSACTION%.

   This synonym contains the action statement that is used for all messages in a Begin-End block that SA z/OS does not trigger any action for. The default, NULL, is that no action will be taken and the message does not continue to search for further matches in the same AT.

   See “Generic Synonyms: AOFMSGSY” in *IBM Tivoli System Automation for z/OS Customizing and Programming* for a description of these synonyms.

**NetView Automation Tables**

If you need to build NetView Automation Tables (ATs) in a way that is not supported by the customization dialog, you can use the INGMSGU1 fragment for user entries. INGMSGU1 is included before INGMSG02. You can also use the INGMSGU2 fragment for user entries. INGMSGU2 is included after INGMSG02.

If you want to have additional entries that are only valid to your environment, you can use either a separate AT (specified in the customization dialog) or use one of the user includes. The following shows the AT structure:

```
INGMSG01
    │
    ├── %INCLUDE AOFMSGSY
    │
    ├── %INCLUDE INGMSGU1
    │
    ├── %INCLUDE INGMSG02
    │
    └── %INCLUDE INGMSGU2
```

**INGXINIT**

The communication DST initialization processing will read data that is specified in the DSIPARM member INGXINIT. Copy and edit the

INGXINIT member, which resides in ING.SINGNPRM. Uncomment the following parameters and specify your values:

**GRPID**
2-byte XCF group ID. Default is blank.

**MQM** 4-byte MQ manager name. Use this parameter only if you run MQSeries® for communication. If you use XCF, you do not need to specify MQM.

**DIAGDUPMSG**
This is the number of message buffer IDs that are validated before send and after receive. This is for diagnostic purposes. A value for *nnnnn* may be chosen between 0 (no validation) and 99999. The default is 0 and performance decreases with larger values.

**LIFECYCLE**
This parameter allows you to prepare for Life Cycle Recording in order to debug automation manager-related problems. Normally, SA z/OS Service will advise when Life Cycle Recording should be enabled.

The value of *nnnn* defines the size of the data space in number of megabytes (1 through 2097). A value of 500 is recommended and is sufficient in most situations.

The value of *dataset* specifies the fully-qualified DSN to be used when offloading the dataspace to disk.

**Note:** *nnnn* and *dataset* must be separated by a semicolon without intervening blanks The total length of '*nnnn;dataset*' can be a maximum of 60 bytes.

**LOGSTREAM**
This defines whether or not the NetView agent should establish a connection to the system logger at initialization time. The default is YES. If NO is specified, the following logstreams are not available:

- HSA.WORKITEM.LOG
- HSA.MESSAGE.LOG

**PPI** This needs to be set to YES to establish a connection to the end-to-end automation adapter.

**PPIBQL**
The number of elements in the PPI queue—this indicates how large the response to a request may be. It should be greater than the number of queue elements that you expect to be returned. The default is 3000.

All input requests flow into the PPI queue, so the buffer queue limit, PPIBQL, should match this. If this limit is exceeded (that is, the queue limit is too small):

- The automation adapter might not be able to send any further requests to the SA z/OS agent, and the agent issues a JNI exception with return code 1735:

```
INGX9820E JNI function ingjppi failed with return code 1735.
```

- The SA z/OS agent might not be able to send any responses to the automation adapter, and an AOF350E message is issued.

If you receive these error messages, increase the buffer queue limit.

Requests are lost, but the end-to-end automation operator will receive exception reports. For more details see *IBM Tivoli System Automation for z/OS End-to-End Automation Adapter*.

All parameter values must match with the respective parameters in the PARMLIB member HSAPRM*xx* of the automation manager.

You can specify a GRPID to indicate that a subset of the members of an actual z/OS sysplex is defined in a sysplex group. If specified, the ID may contain 1 or 2 characters. Valid characters are A–Z, 0–9, and the national characters ($, # and @).

The GRPID is prefixed with the string INGXSG to construct the XCF group name that is used for cross system synchronization, for example, INGXSG*xy*.

If you do not specify a GRPID, the default group name INGXSG is used.

**Note:**
Syntax errors are reported by a message with error code ERRCODE=564. Any syntax errors will stop the initialization process and therefore no automation will be possible.

The following parsing syntax applies:

- Data can only be specified via key-value-pairs.
- One or more parameters may be specified on one line.
- Each record will be parsed for the keyword.
- Parsing will be stopped and any further input data will be ignored after all keywords listed above are found.
- If the same parameter is specified multiple times, the last one is used.
- For any keyword that was not specified, the default value is blank.
- No blanks between parameters and values are allowed.
- The syntax of a keyword is equal to the syntax of the parmlib member HSAPRM*xx*.

An example of a valid syntax is:

```
GRPID=XY,MQM=SSSS,LOGSTREAM=YES
```

An example of an invalid syntax is:

```
GRPID = 34 , MQM = SSSS
```

**DSICMSYS/AOFCMDSO/INGCMD**

If you want to use the SA z/OS SETTIMER command instead of the NetView SETTIMER command, use the following:

- **For NetView V5.1:** Use the following in the NetView style sheet:

```
auxInitCmd.AA=ADDCMD NAME=EZLE600A,MOD=DSICCP,
 ECHO=N,CMDSYN=(TIMER,TIMERS,TIMR),REPLACE=Y
auxInitCmd.AB=ADDCMD NAME=AOFRAATA,MOD=DSICCP,
 CMDSYN=SETTIMER,REPLACE=Y
```

- **For NetView V5.2 and higher:** Use the following in the CNMCMDU member:

```
CMDDEF.EZLE600A.CMDSYN=TIMER,TIMERS,TIMR
CMDDEF.AOFRAATA.CMDSYN=SETTIMER
```

## Step 7C: Modifying NetView DSIPARM Definitions for an Automation Network

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | |

**Note:** The following information refers to setting up a single NetView automation network.

To support an automation network, you need to add or modify NetView definitions in the NetView DSIPARM data set member AOFOPFGW.

### AOFOPFGW Modifications

In the AOFOPFGW member for each system, define the operator IDs used for both outbound and inbound gateway autotasks.

For example, in Figure 12 on page 56, the gateway autotask definitions in AOFOPFGW on system CHI01 are:

```
GATCHI01  OPERATOR
          PROFILEN AOFPRFAO
GATATL01  OPERATOR
          PROFILEN AOFPRFAO
GATATL02  OPERATOR
          PROFILEN AOFPRFAO
```

## Step 7D: Customize NetView for Processor Operations

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| | ✓ | |

To enable SA z/OS, make sure that the following TOWER statements are activated in the NetView style sheet:

```
TOWER =  SA
TOWER.SA  = SYSOPS PROCOPS
```

For SNMP and BCP internal interface connections, it is mandatory to make the security definitions described in “Controlling Access to the Processor Hardware Functions” on page 178.

Processor operations uses automation table entries for its operation. Make sure that the following automation table fragments are included in its master members:

**ISQMSG01**
Processor operations requires the automation table ISQMSG01 for its operation. This table is automatically activated when processor operations is started and deactived, once it is stopped. This automation table uses symbols defined in AOFMSGSY. Make sure this automation table contains valid definitions for the variables %AOFOPMSU% and %AOFOPNETOPER%, and that it is accessible at processor operations start time.

**ISQMSGU1**
This empty member is supplied by processor operations and is included in the ISQMSG01 automation table. By inserting your own automation entries or include statements of your own automation tables here, you can expand

processor operations with your own automation routines which may utilize the processor operations supplied command API.

**NetView Style Sheet**

If you have defined target hardware in your processor operations configurations with an SNA based NetView connection (NVC), it is recommended to set the VTAMCP statement in your processor operations focal point NetView style sheet to `VTAMCP=NO`. This is the default when no VTAMCP statement is defined. Depending on your NetView MS environment, messages and alerts from the CPC support elements cannot be processed when the VTAMCP parameter is set to YES.

## Step 7E: Customize the NetView Message Translation Table

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| * | | |

If you use Kanji support, the NetView Message Translation Table that was specified in the NetView style sheet with the `transMember` entry needs to be customized. (The NetView default for the Message Translation Table is CNMTRMSG located in library SDSIMSG1.)

Verify that in the CNMTRMSG member the INCLUDE for CNMMSJPN is uncommented:

```
%INCLUDE CNMMSJPN
```

In addition add includes for the SA z/OS Kanji message members at the beginning of CNMTRMSG:

```
%INCLUDE AOFJ
%INCLUDE EVEJ
%INCLUDE EVIJ
%INCLUDE EVJJ
%INCLUDE INGJ
%INCLUDE ISQJ
```

Note that only the fixed text of the messages has been translated. Any variables inserted into the text cannot be translated using NetView services, even if the variable contains text strings that are in principle translatable.

## Step 7F: Add the INGRXFPG REXX Function Package

SA z/OS has its own REXX function package, INGRXFPG, that must be made known to NetView. Add it to the function package table in the NetView module DSIRXPRM. Refer to the CNMSJM11 sample for the default NetView DSIRXPRM module that includes the function package table, and modify it.

Add the INGRXFPG package to the NetView system function packages as follows:

```
*                                     PACKTB entry
PACKTB_SYSTEM_TOTAL DC F'2'         Total number of SYSTEM PACKTB
*                                     entries
PACKTB_SYSTEM_USED  DC F'2'         Number of used SYSTEM PACKTB
*                                     entries
```

Remember to update the total number of system packages and user function packages accordingly.

Place the resultant data set that contains the function package in the Linklist to gain performance improvements.

## Step 8: Preparing the Hardware

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | |

The steps described in this section are necessary to prepare your Hardware Management Console (HMC) and Support Elements according to the processor hardware interface you are using. For details about planning the hardware interface, refer to "Planning the Hardware Interfaces" on page 27.

In addition, refer to the publications *Hardware Management Console Guide* and *Support Element Operations Guide* for details about your HMC and SE.

The following customization information addresses different versions of the SE or HMC Console Workplace. You can identify the Console Workplace version of your HMC or SE in its main window title line. Choose the installation step that applies to your Console Workplace version.

### Step 8A: Preparing the HMC (Console Workplace 2.8 and Prior Versions)

#### Enable the HMC API and Set the Community Name

In order to control a CPC using an HMC instead of the CPC's Support Element, the Hardware Management Console API function must be enabled. If you do not plan to use the HMC to control your CPCs over the TCP/IP SNMP ProcOps interface, omit this paragraph.

1. For this task, you need to be logged on in *Access Administrator* mode on your HMC.
2. Select **Console Actions** and click on the **Hardware Management Console Settings** icon. On the Settings notebook, note the TCP/IP address of the HMC for later.
3. Select the **API** tab. If not already set, enable the API by checking the enable check box.
4. In the **Community name** field, enter a community name you have chosen. Note this community name for later.
5. Finally, select the **Apply** push button to save the changes. The message window shown informs you that the changes made require a restart of the HMC console application in order to become active.

#### BCP Internal Interface

To prepare the master HMC, carry out the following steps:

1. Log on to the HMC in your LAN that is to be used for change management operations with a user ID having *SYSPROG* authority. The HMC must have the CPC objects of your sysplex in its Defined CPCs Group.
2. Select **Console Actions** icon in the Views window and double click on the **Enable Hardware Management Console Services** icon.
3. Select the LIC Change **Enabled** radio button. Select the **OK** push button to save the change, or select the **Cancel** push button if **LIC Change** radio button was already set to **Enabled**.

Usually, there is one HMC in a CPC LAN environment that has LIC change permanently enabled. It will automatically be used by the BCP internal interface. Make sure that this HMC has all CPC objects of your sysplex in its Defined CPCs Group.

## SNMP

If you want to control your CPCs with the TCP/IP SNMP interface of ProcOps over an HMC, make sure its API is enabled as described in “Enable the HMC API and Set the Community Name” on page 97. Then, continue as follows:

1. Log on to the HMC in *Access Administrator* mode.
2. From the *Console Actions Work Area*, select **SNMP Configuration**.
3. Select the **Communities** tab of the SNMP Configuration notebook window.
4. For the **API** community name, enter the following information and select the **Add** push button to add the new community name:

   **Protocol** Select UDP from the drop-down list.

   **Name** The API Community name you have chosen.

   **Address** The TCP/IP address of the Support Element which you previously made a note of.

   **Network Mask** 255.255.255.255

   **Access Type** Select the **Read only** radio button.

   If the HMC has multiple network adapters, the SNMP API must be defined to use adapter 0 (primary network adapter) even if that adapter is not later being used for network connection.
5. For the **processor operations SNMP interface** community name, enter the information below and select the **Add** push button to add the new community name.

   The CPC is controlled over the TCP/IP SNMP transport if it is configured for connection protocol SNMP, using the Processor (CPC) entry in the SA z/OS Customization Dialog.

   **Protocol** Select UDP from the drop-down list.

   **Name** PROCOPS (Use the community name specified in the processor entry for the CPC in your SA z/OS policy database.)

   **Address** Use the IP address of your MVS processor operations focal point system.

   **Network Mask** Use 255.255.255.255 to make sure that only the addressed focal point can control the CPC. You may change the netmask to allow multiple focal point systems to control your CPC. Specify 0.0.0.0 as the address and network mask if you want to allow access from any location in your network to your CPC, using the community name from above.

   **Access Type** Select the **Read/write** radio button.
6. Select the **OK** push button to save the changed settings and close the SNMP notebook window.
7. If any of the above data was added or changed, you need to shutdown and restart the Console before the changes will be put into effect.

## NVC

The following setup step is required if you want to define an SNA-based NetView connection between the Support Element of a CPC and the processor operations focal point. The CPCs that you want to connect to the processor operations focal point must have a valid SNA address configured and must be in the Defined CPCs Group of the HMC. If the CPC object is not defined to the HMC Defined CPC Group, see the HMC Object Definition task. To complete this setup:

1. Log on to the HMC with a user ID having *SYSPROG* authority.
2. From the selectable *Task List*, choose the **CPC Remote Customization** task.
3. From the *Groups View Work Area*, select the **Defined CPC Group**.
4. Mark the CPC objects you want to select for processor operations focal point communication and click on the **Problem Management** task icon in the *CPC Remote Customization task area*.
5. In the *Problem Management* window that is displayed, select the **Enable alert generation** push button.
6. In the **Focal Point Addressing, LAN address** field, enter the 12-digit LAN address of the device serving as the gateway between the processor hardware LAN and your SNA network. This device, usually a network control unit such as a 37*xx* or 2216, must be physically connected to the processor hardware LAN.
7. Select the **Save** push button to complete the task for Problem Determination.
8. Now select **Operations Management** task and repeat these steps for the CPC objects that are still marked.

## HMC Object Definition

Depending on the processor hardware interfaces, the CPCs that are to be managed must be known by the HMC, used to route OCF requests to other SEs (BCP internal interface), or to the HMC serving as the single point of control (SNMP), or to the HMC that routes alerts from the CPCs to the processor operations focal point (NVC).

Use the following steps to define a CPC object to an HMC:

1. Log on to the HMC with a user ID having *ACSADMIN* authority.
2. From the task list choose the **Object Definition** task. From the *Groups View* select the **Undefined CPC** group.
3. If the CPC object that you want to define to the HMC is shown in the *Undefined CPC's Work Area*, highlight it and then double click on the **Add Object Definition** task in the *Object Definition* tasks window
4. The *CPC Definition Information Notebox* is displayed, showing the available address information for this CPC object. If you do not want to change any of the address information fields or radio button settings, select the **Save** push button. For more information about the address fields or radio buttons, refer to the HMC online help
5. The CPC is now defined to the HMC. The CPC's Support Element is rebooted to activate its registration to this HMC.
6. If the CPC object that you want to define to the HMC is *not* shown in the *Undefined CPC's Work Area*, highlight the **CPC Manual Definition Template** object .
7. The *Manual Add Object Definition* window is displayed. According to your environment, choose which protocol to use for communication between the CPC's Support Element and this HMC.

8. Depending on your protocol selection, enter: An IP address; Or the SNA Network ID and CPC name; Or the token ring address of the LAN bridge in the case of an SNA connection between the HMC and the CPC over a bridged LAN.
9. Select the **OK** push button. The HMC starts to communicate with the CPC using your network information. If the Add was successful, the CPC object will be shown in the *Defined CPCs Work Area*.

## STEP 8B: Preparing the HMC (Console Workplace 2.9 and Later Versions)

### Enable the HMC API and Set SNMP Community Names

In order to control a CPC using an HMC instead of the CPC's Support Element, the Hardware Management Console API function must be enabled. If you do not plan to use an HMC to control your CPCs over the TCP/IP SNMP ProcOps interface, omit this task. To complete this task:

1. For this task, you need to be logged on in *Access Administrator* mode on your HMC.
2. Select **Console Actions** and click on the **Hardware Management Console Settings** icon.
3. Click on the **Customize API Settings** icon. Make sure the **Enable SNMP APIs** check box is set in the Customize API Settings window.
4. **Important:** The window field SNMP agent parameters must be empty. Any data in this field will prevent the console application from establishing an API session successfully.

For SNMP connections to the HMC, the community names must be defined. After that, you can use native SNMP commands to query and set HMC object attributes, or you can use SA z/OS ProcOps to manage CPCs defined on the HMC and to execute CPC HW commands over the SA z/OS ProcOps SNMP interface.

In SA z/OS, a CPC is controlled over the SNMP interface if it is configured for connection protocol SNMP, using the Processor (CPC) entry in the SA z/OS Customization Dialog.

5. The Customize API Settings window must be open. For a new ProcOps SNMP interface community name, select the Community Names table **Add** push button. In the Community Name data entry window enter the following information:

   **Name**
   Use the community name specified in the processor entry for the CPC in your SA z/OS policy database for ProcOps.

   **Address**
   Use the IP address of your SA z/OS ProcOps focal point system.

   **Network Mask**
   Use 255.255.255.255 to make sure that only the addressed focal point can control the CPC.

   You may change the netmask to allow multiple focal point systems to control your CPC with the same community name. Specify 0.0.0.0 as the address and network mask if you want to allow access from any location in your network to your CPC, using the community name defined.

| | |
|---|---|
| **Access Type** | Select the **Read/write** radio button. |

6. Select the **OK** push button to save the changed settings and close the data entry window.
7. If you have finished the SNMP API settings, select the **Apply** push button of the Customize API Settings window to save the changes.
8. The SNMP Configuration Info window is displayed to inform you that the HMC console must be restarted to activate your configuration changes.

### BCP Internal Interface

To prepare the master HMC, carry out the following steps:

1. Log on to the HMC in your LAN that is to be used for change management operations with a user ID having *SYSPROG* or *ACSADMIN* authority. The HMC must have the CPC objects of your sysplex in its Defined CPCs Group.
2. Select **Console Actions** and click on the **Hardware Management Console Services** icon.
3. Select the **Customize Console Services** icon.
4. Make sure the **LIC Change** field in the Console Services window is set to **Enabled**.
5. Select the **OK** push button to save the change, or the **Cancel** push button if the **LIC Change** radio button was already set to **Enabled**.

Usually, there is one HMC in a CPC LAN environment that has LIC change permanently enabled. It will automatically be used by the BCP internal interface. Make sure that this HMC has all CPC objects of your sysplex in its Defined CPCs Group.

### CPC Object Definitions on the HMC

Depending on the processor hardware interfaces, the CPCs that are to be managed must be defined to the HMC. For SA z/OS's BCP internal interface, the master HMC, which must have the 'LIC Change' service enabled, is used as a router between the CPC where SA z/OS is running, and other targeted CPCs.

For SA z/OS's ProcOps SNMP connection, the HMC serves as a single point of control. Alternatively, SA z/OS ProcOps can be configured to communicate directly with a CPC, by addressing its Support Element.

For detailed information on how to add, change, or remove CPC object definitions on a HMC, refer to the current *Hardware Management Console Operations Guide* (SC28-6821). Note that this manual is also available in the Books Work Area on the HMC.

## Step 8C: Preparing the SE (Console Workplace 2.8 and Prior Versions)

Before the BCP internal interface can be used, you need to verify for the CPC Support Elements in your sysplex that the required prerequisite MCL levels are active, and that any essential services have been enabled with the necessary settings. This requires the following:

- “Configure SNMP” on page 102
- “Enable the API and Set the Community Name” on page 103
- “Set the Cross Partition Flags” on page 103 (LPAR mode)
- “Customize the Authorization Token” on page 103

## Configure SNMP

Community names have to be specified in order to use the BCP internal interface transport, the TCP/IP SNMP transport for ProcOps, or both. For this task, you need to be logged on in *Access Administrator* mode on your CPC's Support Element. To complete this task:

1. Start the SNMP Configuration task by double clicking the **Console Actions** icon in the *Views* area of the Console.
2. Select the **Communities** tab of the SNMP Configuration notebook window.
3. For the **API** community name, enter the following information and select the **Add** push button to add the new community name:

   | | |
   |---|---|
   | **Protocol** | Select UDP from the drop-down list. |
   | **Name** | The API Community name you have chosen. |
   | **Address** | The TCP/IP address of the Support Element which you previously made a note of. |
   | **Network Mask** | 255.255.255.255 |
   | **Access Type** | Select the **Read only** radio button. |

   If the SE has multiple network adapters, the SNMP API must be defined to use adapter 0 (primary network adapter) even if that adapter is not later being used for network connection.
4. If the CPC is not controlled over the BCP internal interface transport, omit this step.

   The CPC is controlled over the BCP internal interface if it is configured for connection protocol INTERNAL, using the Processor (CPC) entry of the SA z/OS Customization Dialog.

   For the **BCP Internal Interface** community name, enter the following information and select the **Add** push button to add the new community name:

   | | |
   |---|---|
   | **Protocol** | Select UDP from the drop-down list. |
   | **Name** | SAFOS (Use the CPC authtkn name that you defined for the CPC using the customization dialogs) |
   | **Address** | 127.0.0.1 |
   | **Network Mask** | 255.255.255.255 |
   | **Access Type** | Select the **Read/write** radio button. |
5. If the CPC is not controlled over the ProcOps TCP/IP SNMP transport, omit this step.

   The CPC is controlled over the TCP/IP SNMP transport if it is configured for connection protocol SNMP, using the Processor (CPC) entry of the SA z/OS Customization Dialog.

   For the **ProcOps SNMP interface** community name, enter the following information and select the **Add** push button to add the new community name:

   | | |
   |---|---|
   | **Protocol** | Select UDP from the drop-down list. |
   | **Name** | PROCOPS (Use the community name specified in the processor entry for the CPC in your SA z/OS policy database.) |
   | **Address** | *x.x.x.x* (Use the IP address of your MVS ProcOps focal point system.) |

| | |
|---|---|
| **Network Mask** | *x.x.x.x* (Use **255.255.255.255** to make sure that only the addressed focal point can control the CPC. You may change the netmask to allow multiple focal point systems to control your CPC. Specify **0.0.0.0** as both the address and network mask if you want to allow access from any location in your network to your CPC, using the community name from above.) |
| **Access Type** | Select the **Read/write** radio button. |

6. Select the **OK** push button to save the changed settings and close the SNMP notebook window.
7. If any of the above data was added or changed, you need to shutdown and restart the Console before the changes will be put into effect. However, before doing so, continue with the configuration steps for Console below.
8. If SNMP configuration data was added or changed, you need to reboot the Support Element to activate these changes.

For additional SNMP and API configuration information, refer to chapter "Configuring the Data Exchange APIs" in *zSeries 900 Application Programming Interface*.

## Enable the API and Set the Community Name

In order to use the BCP internal interface or the SNMP interface, the Support Element API function needs to be enabled. To complete this task:

1. Start the Support Element Settings task by double clicking the **Console Actions** icon in the *Views* area of the Console.
2. Select the **API** tab of the Support Element Settings notebook window. If not already active, enable the API by checking the **Enable the Support Element Console Application Program Interface** checkbox.
3. In the **Community name** field, enter the community name you chose when you configured for SNMP.
4. Select the **Apply** push button to save the changes.
5. Finally, for the changes you have made to the Support Element to become active, you must reboot the Support Element.

## Set the Cross Partition Flags

This task is only required if you use the BCP internal interface to connect processor hardware running in LPAR mode. For this task, you need to be logged on in *System Programmer mode* on your CPC's Support Element. To complete this task:

1. Click on the **CPC Group** and highlight the **CPC** icon.
2. Select the **CPC Operation Customization** task.
3. Click on the **Change LPAR Security** icon. The window displayed shows the security settings from the active IOCDS for the logical partitions defined on this CPC.
4. For each logical partition that should use the BCP internal interface to control another partition on this CPC, check the **Cross Partition Authority** checkbox.

## Customize the Authorization Token

This task is only required for NVC connections. To complete this task:

1. Log on to the Support Element with a user ID having *ACSADMIN* authority.
2. Select the **Console Actions View**. The *Console Action Work Area* is displayed.

3. Double click on the **Customize Authorization Token** icon. If you want to change the supplied default, enter a new authorization token value. The new value can be up to 8 characters long and must not contain blanks. This authorization token value must be used when defining a NVC for a CPC using the SA z/OS customization dialogs.

### Additional Verification for SEs of G3/G4 CPC Hardware

When customizing the Support Element for a G3/G4 machine, it should be verified that the SETUP.CMD file in the Support Element subdirectory C:\MPTN\BIN contains the statement `'ifconfig lo 127.0.0.1'`. Only if this statement is defined, can the BCP internal interface be used to target this machine.

If, after having made the necessary HMC and SE SNMP definition steps for a G3/G4 CPC, the BCP internal interface communication to this hardware still fails with a condition of 0B100224, contact your IBM customer engineer to perform the above-mentioned verification. Should the communication still fail, contact IBM software service.

## 8D: Preparing the SE (Console Workplace 2.9 and Later Versions)

### Enable the SE API and Set the Community Name

To control a CPC with the SA z/OS hardware interfaces BCPii or SNMP ProcOps directly, the CPC Support Element API function must be enabled. To complete this task:

1. For this task, you need to be logged on in *Access Administrator* mode on your HMC.
2. Select **Console Actions** and click on the **Support Element Settings** icon.
3. Click on the **Customize API Settings** icon. Make sure the **Enable SNMP APIs** check box is set in the Customize API Settings window.
4. **Important:** The window field SNMP agent parameters must be empty. Any data in this field will prevent the console application from establishing an API session successfully.

**Set the Community Name for SNMP and ProcOps Connections.**

For SNMP connections to the SE, the community names must be defined. After that, you can use native SNMP commands to query and set SE object attributes, or you can use SA z/OS ProcOps to manage the CPC and to execute CPC HW commands using the SA z/OS ProcOps SNMP interface.

In SA z/OS, a CPC is controlled over the SNMP interface if it is configured with connection protocol SNMP in the Processor (CPC) entry of the SA z/OS Customization Dialog.

5.
   a. The Customize API Settings window must be open. For a new ProcOps SNMP interface community name, select the Community Names table **Add** push button. In the Community Name data entry window enter the following information:

| | |
|---|---|
| **Name** | Use the community name specified in the processor entry for the CPC in your SA z/OS policy database for ProcOps with connection type SNMP. |

**Address**
Use the IP address of your SA z/OS ProcOps focal point system.

**Network Mask**
Use 255.255.255.255 to make sure that only the addressed focal point can control the CPC.

You may change the netmask to allow multiple focal point systems to control your CPC with the same community name. Specify 0.0.0.0 as the address and network mask if you want to allow access from any location in your network to the SE, using the community name defined.

**Access Type**
Select the **Read/write** radio button.

**Set the Community Name for a BCP Internal Interface Connection**

For BCPii connections to the SE, a community name must be defined.

In SA z/OS, a CPC is controlled over the BCPii if it is configured with a connection protocol INTERNAL, using the Processor (CPC) entry of the SA z/OS Customization Dialog.

b. The Customize API Settings window must be open. For a new BCP internal interface community name, select the Community Names table **Add** push button. In the Community Name data entry window enter the following information:

**Name**
Use the community name specified in the processor entry for the CPC in your SA z/OS policy database for ProcOps that has the connection type INTERNAL.

**Address**
The required address is 127.0.0.1

**Network Mask**
The required value is 255.255.255.255

**Access Type**
Select the **Read/write** radio button.

6. Select the **OK** push button to save the changed settings and close the data entry window.
7. If you have finished the API settings, select the **Apply** push button of the Customize API Settings window to save the changes.
8. The SNMP Configuration Info window is displayed to inform you that the SE console must be restarted to activate your configuration changes.

**Set the Cross Partition Flags:** This task is only required if you use the BCP internal interface to connect processor hardware running in LPAR mode. For this task, you need to be logged on in *System Programmer mode* on your CPC's Support Element. To complete this task:

1. Click on the **CPC Group** and highlight the **CPC** icon.
2. Select the **CPC Operation Customization** task.
3. Click on the **Change LPAR Security** icon. The window displayed shows the security settings from the active IOCDS for the logical partitions defined on this CPC.
4. For each logical partition that should use the BCP internal interface to control another partition on this CPC, check the **Cross Partition Authority** checkbox.

## Step 8E: Updating Firewall Information

This step is only needed if you use ProcOps and intend to use TCP/IP based communication to your target processors.

### Connection protocol SNMP

This communication protocol internally uses port number 3161. If there are firewalls installed between the LAN that the ProcOps FP belongs to and the processor LAN that the SEs or HMCs belong to, you should:

- Inform your network administrator to make sure that communication requests that come from SEs/HMCs with this port number are accepted.

# Step 9: Preparing the VM PSM

| SysOps | ProcOps | I/O Ops |
|---|---|---|
|  | * |  |

This step is only needed if you use ProcOps to control VM second level systems. The PSM is the communication partner for ProcOps to do this.

## Installing the PSM Code on VM

The following parts are shipped as part of the Second Level Guest Support feature:

- In *xxx*. SINGOBJV — module ISQVMAIN (this is the PSM control program's main thread)
- In *xxx*.SINGREXV the following squished REXX programs:
  - ISQRGIUC
  - ISQRCSRV
  - ISQRMSRV
  - ISQRLOGR
  - ISQRCNSV
  - ISQRMHDL
- In *xxx*.SINGMSGV — Message definitions ISQUME

To install the VM parts perform the following steps:

1. Copy the object module ISQVMAIN to the VM file system for the PSM machine as file `ISQVMAIN TEXT`
2. Copy REXX programs to the VM file system for the PSM machine as files:
   - `ISQRGIUC REXX`
   - `ISQRCSRV EXEC`
   - `ISQRMSRV EXEC`
   - `ISQRLOGR EXEC`
   - `ISQRCNSV EXEC`
   - `ISQRMHDL EXEC`
3. Copy message definition ISQUME to the VM file system for the PSM machine as file `ISQUME REPOS`
4. Enter the following commands on the PSM machine (These may be created as an CMS EXEC if necessary). The name chosen for the operand of the GENMOD command (ISQPSM in this case) defines the name of the PSM control program. Any name may be chosen. These commands create the load module for the PSM main thread and the messages definitions for all threads.

```
GENMSG ISQUME REPOS A ISQ
SET LANG (ADD ISQ USER
GLOBAL TXTLIB DMSAMT VMMTLIB VMLIB
LOAD ISQVMAIN
INCLUDE ISQUME
INCLUDE VMSTART (LIBE RESET VMSTART
GENMOD ISQPSM
```

5. Create the two files ISQADDRS DATA and ISQPARM DATA as described in "Customizing the PSM" on page 108.

If these steps are processed successfully then the PSM can be started.

## Configuration

1. Provide TCPIP connection between the VM host system and the SA z/OS systems that are running NetView ProcOps.
2. Define a ProcOps Service Machine in each VM host. This is a regular virtual machine that IPLs a CMS when it starts. Ensure that it has a minimum of 32 MB of storage defined.
3. Use the IUCV directory control statement to authorize the PSM virtual machine to connect to the CP message service (*MSG). For more information about the IUCV statement, see the *z/VM: Planning and Administration* book.
4. Authorize the ProcOps Service Machine to use CP and CMS commands. The following commands are used by the PSM:

   ```
   SET SECUSER vmachine *
   SET EMSG
   TERMINAL MORE
   SET VMCONIO
   SET CPCONIO
   GLOBALV
   XAUTOLOG
   FORCE
   XMITMSG
   SEND
   SMSG
   QUERY NAMES
   QUERY vmachine
   ```

5. Optionally, ensure that the language is set automatically and that the ProcOps Service Machine starts when the PSM virtual machine starts by creating a PROFILE EXEC for the virtual machine (if one does not already exist) and adding the appropriate commands to it:

   ```
   SET LANG (ADD ISQ USR
   ISQPSM
   ```

   where ISQPSM is the name of the control program in the earlier example.
6. Ensure that the ProcOps Service Machine has appropriate dispatching priority. Ideally it should have a higher dispatching priority than the guest machines that it manages.
7. Define the PSM as a Service Virtual Machine.
8. For each guest machine, ensure that the PSM virtual machine is defined as its secondary user
9. Define SYSCONS as a NIP console and MCS console for each guest MVS machine, with appropriate routing codes
10. It is recommended that the PSM virtual machine has read access to the minidisk that holds the TCPIP program, so that the NETSTAT command can be issued as part of problem determination procedures.

## Customizing the PSM

The PSM uses two files to set parameters for its operation. These files are read at the time that PSM is initialized, and are not read subsequently.

The statements in them determine the various operational characteristics.

Each file is a simple sequential file that must be part of the file system available to the PSM virtual machine. Normally they are files on the A-disk. Each file must be available at PSM initialization. If any is missing, the PSM terminates.

### ISQADDRS DATA

The ISQADDRS DATA file specifies those IP addresses that may enter requests to the PSM. Each ProcOps NetView that issues requests to the PSM must have its IP address specified.

Each record of the file specifies a single IP address. Any record that has an asterisk in the first position is treated as a comment. Any record that has the string ″/*″ in the first two positions is treated as a comment.

The IP address may be specified either in the normal dotted decimal form, or as a node name that is known to TCPIP on the PSM's node. If a node name is specified and that node name has several addresses, all addresses that are returned are used.

An example of a valid file is as follows:

```
*  Normal focal point NetView
9.152.80.253
/*  the backup
  9.152.80.254
* another system identified by its node name
  nv.boekey3.de.ibm.com
* a shorter, if infrequent form of IP address
44.55
```

The addresses are *not* checked for validity when they are read.

### ISQPARM DATA

The ISQPARM DATA file specifies operational options for the PSM.

Each record of the file specifies a single parameter. Any record that has an asterisk in the first position is treated as a comment. Any record that has the string ″/*″ in the first two positions is treated as a comment.

The statements are of the form:

```
keyword = value
```

All keywords, except TCPIPNAME, must be specified. If any required keywords are omitted the PSM will terminate. The keywords may be entered in upper, lower or mixed case. Values must be entered as required. If a keyword specification is entered more than once, the latest specification is used.

Valid keywords are:

**MESSAGE_SERVER_PORT**
The port number that will be used by the Message Server. (That is, the port on which it issues a TCPIP LISTEN request.) This is a number in the range 1-65335. Consult with your network programmer to ensure that this is a port number that is not used by any other processes.

**COMMAND_SERVER_PORT**
The port number that will be used by the Command Server.

**SECURITY**
The authorization token used to authenticate both the Message Server and Command Server. This must match the authorization token that is specified in the System Automation Customization dialogs for this PSM Target Hardware. This must have the correct (upper) case.

**TCPIPNAME**
The name of the TCPIP virtual machine that will provide the connections to ProcOps NetView. When the PSM control program starts, it checks that this virtual machine is running before issuing any TCPIP requests. The default value used, if TCPIPNAME is not specified, is TCPIP.

**MAX_MESSAGES**
The maximum number of messages that may be stored at any instant in the Message Queue. When the number of messages in the queue exceeds this number, the Message Handler thread terminates with an error message.

**TRACE_TYPE**
The trace type identifies the trace type value that is entered into log records written by the Logger thread.

An example of a valid file is:

```
Message_server_port = 5556
Command_server_port = 4444
*
TRACE_TYPe = 555
security = ISQHELLO
max_messages = 20
```

### Logger Files

The PSM must also have sufficient writeable space on its A-disk to accommodate the logger files and any files that might be used by CP commands such as DUMP, if used.

## Step 10: Customizing the Automation Manager

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | | |

### Step 10A: Customizing HSAPRM*xx*

The HSAPRM*xx* PARMLIB member contains information required for the initialization of the automation manager and default values for other operational parameters. The member is designed to be used in common by all automation manager instances in the automation subplex.

Alternatively you can put the automation manager PARMLIB member in any partitioned data set. Then, you need to insert a statement HSAPLIB DD into the automation manager startup procedure member which refers to this partitioned data set.

A sample member called HSAPRM00 is provided in the SINGSAMP sample library. This sample is automatically copied into the PARMLIB of the automation manager (DD name HSAPLIB) when you allocate this data set as described in

"Step 2: Allocate System-Unique Data Sets" on page 76. Refer to Appendix G, "Syntax for HSAPRM00," on page 247 for the contents of this sample and the description of the parameters.

## Step 10B: ARM Instrumentation of the Automation Manager

The automation manager can be enabled for Automatic Restart Manager (ARM).

A job skeleton is provided in the SINGSAMP sample library as member HSADEFA to define the SA z/OS specific Automatic Restart Manager policy.

You can define a policy allowing you to keep the number of automation manager instances on a certain level.

**In a single system environment**

With more than one automation manager active, ARM can automatically restart a failing primary instance. One of the automation managers that survived will take the primary role and the restarted instance will become a backup instance.

If there is only one automation manager active on a single system, ARM will automatically restart this instance again. It becomes the primary instance again and runs the takeover. The takeover time is extended by the time needed for the address space restart.

**In a sysplex (subplex) environment**

ARM will always restart the failing instance on the **same** system. Either there is already a backup waiting or the restarted instance will take over.

SA z/OS provides a policy sample with the following major options:

- Restart only for an address space ABEND (Option ELEMTERM). Restart in case of a system breakage is not supported.

  The concept of the automation manager availability follows a 'floating' master model. It is a peer model with one or more backup instances on different systems already active and waiting to take over. Whenever a complete system goes away the failed automation managers (backup or primary) are not restarted somewhere else.
- The ARM element name is a 16 byte string concatenation `HSAAM_sysnamexy` with:

  **HSAAM_**
  is a string constant as prefix

  **sysname**
  Is the XCF member name of the automation manager which is the 8 byte MVS system name padded with '$', for example, MVS1$$$$

  **x** Is a one byte digit (one of 1, 2, ... 9) automatically determined at initialization time

  **y** Is a blank
- The restart command is the unchanged original start command, however the start mode is always HOT.
- There are no restart dependencies (no Waitpred processing)

## Step 10C: Security Considerations

The job invoking the automation manager (see INGEAMSA in the sample library) must have the following access rights:

1. If you are not a superuser you must have access to the OMVS segment.

2. It must be defined by RACF as a superuser for UNIX System Services if the automation manager will be started before JES2 initialization has completed.
3. Read access for the SYS1.PARMLIB data set
4. Write access to the log streams
5. Write access to the following data sets:
   - Trace data sets
   - Schedule override file
   - Configuration information file (DDname HSACFGIN)
   - Takeover file

## Step 11: Customizing the Component Trace

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | | ✓ |

Both the system operations component and the automation manager use the z/OS component trace for debugging purposes. The following setup must be done:

- Copy the CTIHSAZZ member from the SINGSAMP sample library to SYS1.PARMLIB. Do not change this member.
- Copy the CTIIHVZZ member from the SINGSAMP sample library to SYS1.PARMLIB. You may change this member to meet your requirements. Refer to "Appendix C. Problem Determination" in *IBM Tivoli System Automation for z/OS User's Guide* for more information.
- Copy the HSACTWR member residing in the SINGSAMP sample library into SYS1.PROCLIB.
- Allocate the trace data set used by the component trace. You can use the sample job HSAJCTWR in SINGSAMP to allocate the data set. Modify the sample job where appropriate.

**Note:** Make sure that the job invoking the ITTTRCWR module (see HSACTWR member in the sample library) has write access to the trace output data set.

## Step 12: Customizing the System Logger

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| * | | |

**Notes:**

1. If you set the LOGSTREAM parameter in the HSAPRM*xx* parmlib member to NO, no access will be established to the system logger at initialization. This step is then unnecessary.
2. Though this step is optional, it is, however, recommended. The automation manager writes history information to the z/OS system logger and the automation agents read from it.

   If you do not perform this step, users will not get any output from the INGHIST commands.

## Step 12: Customizing the System Logger

To exploit the system logger, the following must be fulfilled:

- Systems in a sysplex must run in XCF mode and the following must be defined in SYS1.PARMLIB(IEASYS*xx*):

  ```
  PLEXCFG=MULTISYSTEM
  ```

- For standalone systems the following must be defined in SYS1.PARMLIB(IEASYS*xx*):

  ```
  PLEXCFG=MONOPLEX
  ```

Next, the LOGR couple data sets must be formatted, if this has not already been done. For this task you can use the sample JCL provided in the HSAJFCDS member of the sample library.

Use the following sample JCLs to define the log stream in different environments:

- For a single system environment, use the sample JCL provided in member HSAJDLGM (for the automation manager)
- For a sysplex, use the sample JCL provided in member HSAJDLGS (for the automation manager)

In both cases you may want to adapt the HLQ parameter in the LOGR policy according to your environment. The default is IXGLOGR. Use the corresponding INGJD*xxx* members as input and make the changes accordingly.

For a sysplex environment, you must additionally add the log structures to the CFRM policy:

```
STRUCTURE   NAME(HSA_LOG)
            SIZE(8192)
            PREFLIST(cfname,cfname)
```

In this CFRM policy, you have to adapt the PREFLIST for structure HSA_LOG if you are setting up the system logger. Also adapt the SIZE parameter to a recommended minimum of 8 megabytes (8M).

The system logger must be authorized. If it is not yet assigned either privileged or trusted RACF status, or both, refer to chapter "Planning for System Logger Applications" in *z/OS MVS Setting Up a Sysplex* for more information on how to define authorization to system logger resources. The names of the system logger resources used by SA z/OS are HSA.MESSAGE.LOG and HSA.WORKITEM.HISTORY.

The address spaces of the NetView agents and automation manager need to be authorized to access the log streams. They need update access for the following:

```
RESOURCE(logstream_name)
CLASS(LOGSTRM)
```

Where *logstream_name* stands for HSA.MESSAGE.LOG and HSA.WORKITEM.HISTORY.

For further information see section "Define Authorization to System Logger Resources" in *z/OS MVS Setting Up a Sysplex*.

Now activate the couple data sets via the console commands:

```
SETXCF COUPLE,TYPE=LOGR,PCOUPLE=(primary_couple_data_set)
SETXCF COUPLE,TYPE=LOGR,ACOUPLE=(alternate_couple_data_set)
```

For a sysplex, after defining the new structure in the CFRM policy, activate the CFRM policy via:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=policy_name
```

## Step 13: Install ISPF Dialog Panels

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | ✓ |

SA z/OS ships two types of ISPF dialogs: one for I/O operations and one for defining automation policy. The I/O operations panels are used for I/O functions. The customization dialog is used to create configuration and automation definitions.

The I/O operations and customization dialog are both invoked using the INGDLG exec. This exec provides parameters for selection of the appropriate dialogs. In addition, this exec can optionally be used to allocate the required dialog libraries. INGDLG should be invoked from an ISPF menu or from a user-defined TSO REXX exec. See Appendix H, "INGDLG Command," on page 255 for more details.

Because you use the customization dialog to collect information and build control files, you normally need them only at the focal point. However, as the customization dialog allows editing of specific entry types by multiple users, you also need to observe the instructions given in the appendix *Problem Determination* in *IBM Tivoli System Automation for z/OS User's Guide*.

The I/O operations dialogs, however, are used to input commands and get responses from the I/O operations part of SA z/OS. Because they do not support multisystem commands for I/O operations functions, you must install them on each system, focal point or target, where you want to use them. Alternatively, you can use the workstation window set to access I/O operations function.

### Step 13A: Allocate Libraries for the Dialogs

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | ✓ |

To set up the dialogs, you must allocate the REXX load libraries and customization dialog load libraries. This section describes the two alternative options available:
- **Alternative 1:** Dynamic allocation of the libraries using the INGDLG exec
- **Alternative 2:** Allocation of the libraries as part of the TSO logon procedure

**Remember:**
Throughout this step use the names of the data sets that you created in "Step 3: Allocate Data Sets for the Customization Dialog" on page 80.

#### Alternative 1: Dynamic Allocation using INGDLG

This exec performs allocations prior to starting the dialogs. In order to invoke the exec, you need to be in ISPF. The INGDLG command parameters describe where the data sets are found. See Figure 30 on page 255 for the use of INGDLG to allocate libraries.

Note that if you use INGDLG to allocate libraries, you must still perform allocation of the ISPF product libraries as described in “Alternative 2: Add to the TSO Logon Procedure.”

## Alternative 2: Add to the TSO Logon Procedure

Create a new TSO logon procedure that has the SA z/OS data sets in the appropriate concatenations.

To create a TSO logon procedure, take an existing one and modify its DD statements to include the following:

```
//ISPPLIB    DD ...
             DD DSN=ING.SINGIPNL,DISP=SHR
             DD ...

//ISPMLIB    DD ...
             DD DSN=ING.SINGIMSG,DISP=SHR
             DD ...

//ISPSLIB    DD ...
             DD DSN=ING.SINGISKL,DISP=SHR
             DD ...

//ISPTLIB    DD ...
             DD DSN=ING.CUSTOM.AOFTABL,DISP=SHR   1
             DD DSN=ING.SINGITBL,DISP=SHR
             DD ...

//ISPLLIB    DD ...
             DD DSN=ING.SINGMOD1,DISP=SHR
             DD ...

//SYSPROC    DD ...
             DD DSN=ING.SINGIREX,DISP=SHR
             DD ...

//AOFTABL    DD DSN=ING.CUSTOM.AOFTABL,DISP=SHR   1

//AOFPRINT   DD SYSOUT=...  2

//AOFIPDB    DD DSN=ING.SINGIPDB,DISP=SHR   3

//IHVCONF    DD DSN=ING.CUSTOM.IHVCONF,DISP=SHR   4
```

**Notes:**

1. Ensure that your ISPF temporary data sets have been allocated with enough space.
   - When a build of the automation control file is performed, each file is written to the temporary data sets before it is copied into the target data set. This can lead to a temporary data set many thousands of lines long. For an enterprise with many processors, there may be several hundred thousand lines written to the temporary data set. These are in the ISPWRK data sets. See *z/OS ISPF Planning and Customizing* for more information, where it is recommended that you pre-allocate to VIO however, because it reduces overhead and eliminates potential problems from insufficient space.
   - The ISPCTL1 temporary data set is used by SA z/OS to hold output created by a data model report and to hold the JCL for batch submission of an ACF Build job. See *z/OS ISPF Planning and Customizing* for more information on the ISPCTL1 data set.
2. Ensure that the ISPF table output library ISPTABL is allocated. The table output data set must also be in the sequence of data sets allocated to ISPTLIB. Furthermore it is recommended that the first data set allocated to ISPTLIB is

user-specific. This is guaranteed if INGDLG is called with the default of ALLOCATE(YES). Then the user's ISPPROF data set is automatically defined as the first data set, and the table output data set is allocated as well. If the first data set allocated to ISPTLIB is not-user specific, multiple users may experience enqueue problems if working with the same PDB concurrently. The reason is that when ISPF opens a table, it requests an enqueue for a resource name that consists of a table name and the first data set allocated to ISPTLIB. For more information, see *z/OS ISPF User's Guide Vol I*.

3. The ellipses (...) in the DD-statements indicate the presence of more information in the JCL: for example, other data sets in a concatenation.
4. User-specific data sets should be placed before the SA z/OS data sets. Generally speaking you need to take care that the concatenation of the SA z/OS data sets does not interfere with the concatenation with data sets from other products.
5. You should **not** include data sets for any predecessor products, AOC/MVS, TSCF, or ESCON Manager, anywhere in the concatenation.
6. The AOFTABL DD statement ( **1** ) is required as soon as you intend to customize your environment: this data set stores ISPF tables containing unique information created when you use the customization dialog. ING.CUSTOM.AOFTABL, allocated in "Step 3: Allocate Data Sets for the Customization Dialog" on page 80, is used to hold new and modified ISPF tables created when the administrator modifies or changes the SA z/OS policy definitions from the SA z/OS customization dialog.Because changes in the SA z/OS policy definitions and ISPF tables might often occur, this makes it a required DD statement. This data set is also used to hold the data set definitions for batch processing. This data set was allocated by you in the sample INGEDLGA (see "Step 3: Allocate Data Sets for the Customization Dialog" on page 80).
7. The AOFPRINT DD statement ( **2** ) is used in place of SYSPRINT for IEBUPDTE, which is invoked when a user of the customization dialog creates a policy database using an SA z/OS-supplied sample as a model. If this DD statement is not allocated, SA z/OS allocates the DD as SYSOUT=H.

   If the IEBUPDTE invocation is successful and SA z/OS dynamically allocated the AOFPRINT file as SYSOUT=H, the output is purged. If the invocation fails, the output is saved for use in diagnosis of the problem.

   When specifying `AOFPRINT(SYSOUT(Cls))`, the output of the dynamically called IEBUPDATE utility is placed in the JES output class *Cls*. This output is not purged.
8. The AOFIPDB DD statement ( **3** ) points to the SA z/OS sample library.

   The AOFIPDB DD statement is required for building system operations control files (automation control file and automation manager configuration file). It must point to a single data set, not a concatenation. In SA z/OS, this data set is required, even if you do not use any sample policy databases. AOFIPDB contains the automation manager logic deck INGLOGIC.
9. IHVCONF ( **4** ), is required for I/O operations. If you are not using I/O operations this DD statement is optional.
10. You should not use any DD names starting with AOF in your logon procedure except those specified in the example above. This is because the SA z/OS customization dialog may dynamically generate AOF*xxxxx* DD names. Specifically, SA z/OS generates AOFIN and AOFUT2 DD names.
11. I/O operations ISPF dialogs use REXX execs that invoke I/O operations commands and ISPF services. These execs must be made available to the users who want to use the ISPF dialogs. Note that the default record format of the

I/O operations REXX target library (whose name is SINGIREX) is FB. The data sets in your SYSPROC concatenation might not be FB. If this is the case, the ALLOCATE command can be used, but you are not able to execute the differently formatted or sized execs. You can do one of the following to correct this:

a. Copy the contents of the SINGIREX exec library to another data set that is already in your SYSPROC concatenation.
b. Copy the contents of the SINGIREX exec library to a new data set that has the same characteristics as the other data sets in your SYSPROC concatenation.

If you already use a CLIST to allocate your data sets for ISPF, modify it to include the SA z/OS data sets in the appropriate concatenations for users of the customization dialog. If you want to create a CLIST to allocate your data sets you should find out your current allocations for the DD names that need SA z/OS data sets allocated to them. This can be done with the LISTALC STATUS command.

## Step 13B: Invoking the ISPF Dialogs

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | ✓ |

The ISPF Application Selection Menu can be modified to include options for the system operations and processor operations customization dialog and for the I/O operations dialogs. These options allow a user to begin the customization dialog without having to issue commands at the TSO prompt.

Two changes are required to add the dialogs to the ISPF Application Selection Menu panel (see also Figure 15 on page 117):

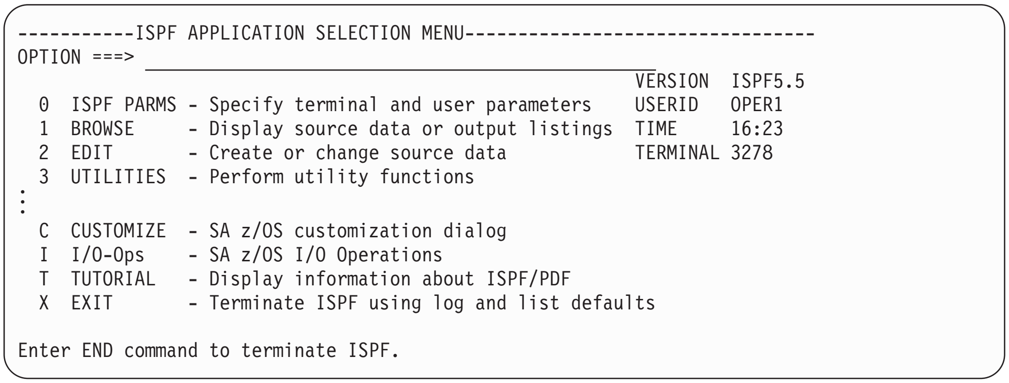
- Adding selections to the menu
- Adding logic to the panel processing to invoke the appropriate dialogs

Both sets of dialogs are invoked by the INGDLG command. Parameters of this command determine which set of dialogs is invoked.

- Add the command dialogs selections to an ISPF menu panel, such as the ISPF Master Application Menu panel (ISP@MSTR) or the ISPF Primary Menu panel (ISP@PRIM).

  **Note:** If you use a customized, non-standard ISPF primary menu panel, modify the definition for that panel instead of ISP@MSTR or ISP@PRIM.

  See *z/OS ISPF Planning and Customizing* for information about customizing ISPF panels. The modified panel should be placed in a data set so that it is used by all users who have the dialog data sets in their concatenation, but it is not used by anyone who does not. You may want to copy it into an enterprise-specific panel data set that you allocate in front of your normal ISPF panel data sets. Figure 15 on page 117 is an example of what a modified panel might look like.



```
-----------ISPF APPLICATION SELECTION MENU---------------------------------
OPTION ===> _________________________________________________
                                                          VERSION  ISPF5.5
  0  ISPF PARMS - Specify terminal and user parameters    USERID   OPER1
  1  BROWSE     - Display source data or output listings  TIME     16:23
  2  EDIT       - Create or change source data            TERMINAL 3278
  3  UTILITIES  - Perform utility functions
⋮
  C  CUSTOMIZE  - SA z/OS customization dialog
  I  I/O-Ops    - SA z/OS I/O Operations
  T  TUTORIAL   - Display information about ISPF/PDF
  X  EXIT       - Terminate ISPF using log and list defaults

Enter END command to terminate ISPF.
```

*Figure 15. ISPF Application Selection Menu*

The options for the customization dialog and the I/O operations dialogs must also be added to the panel processing section of the ISPF Application Selection Menu panel as follows. The lines you add are written in italics in the example. You can select the character used to specify the dialogs on your menu.

## Using TSO Logon or Your CLIST

This is the example to be followed if you allocated the data sets using the TSO logon procedure or using a CLIST of your own.

```
)PROC
&ZQ = &Z
IF (&ZCMD ^= ' ')
&ZQ = TRUNC(&ZCMD,'.')
IF (&ZQ = ' ')
 .MSG = ISRUOOO
&ZSEL = TRANS( &ZQ
0,'PANEL(ISPOPTA)'
⋮
C,'CMD(INGDLG SELECT(ADMIN) ALLOCATE(NO))'
I,'CMD(INGDLG SELECT(IOCONNECT) ALLOCATE(NO))'
T,'PGM(ISPTUTOR) PARM(ISR00000)'
⋮
X,'EXIT'
*,'?' )
&ZTRAIL = .TRAIL
)END
```

## Using INGDLG

If you let INGDLG, described in Figure 30 on page 255, allocate the data sets dynamically prior to starting the dialogs, the following is a sample definition to be added to the ISPF processing section:

```
C,'CMD(EXEC ''ING.SINGIREX(INGDLG)'' +
  ''HLQ(MYHLQ)                       +
    AOFTABL(ING.CUSTOM.AOFTABL)      +
    SELECT(ADMIN)'')'
I,'CMD(EXEC ''ING.SINGIREX(INGDLG)'' +
  ''HLQ(MYHLQ)                       +
    IHVCONF(ING.CUSTOM.IHVCONF)      +
    SELECT(IOCONNECT)'')'
```

Alternatively, you can invoke the dialogs using TSO REXX execs:

```
/* REXX ADMIN */
ADDRESS ISPEXEC "SELECT CMD(EXEC 'ING.SINGIREX(INGDLG)'          " ,
"'HLQ(ING)                     " ,
/* HLQ is the hlq of the SMP/E output data sets */
" AOFTABL(ING.CUSTOM.AOFTABL)    " ,
" SELECT(ADMIN)                 ')"
```

```
/* REXX IOCONNECT */
ADDRESS ISPEXEC "SELECT CMD(EXEC 'ING SINGIREX(INGDLG)'          ",
"'HLQ(ING)                      ",
/* HLQ is the hlq of the SMP/E output data sets */
" IHVCONF(ING.CUSTOM.IHVCONF)    ",
" SELECT(IOCONNECT)            ')"
```

A sample member called INGEDLG is provided in SINGSAMP sample library for invocation of INGDLG with data set allocation done by INGDLG.

## Step 13C: Reconvert I/O Operations Panels

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| | | * |

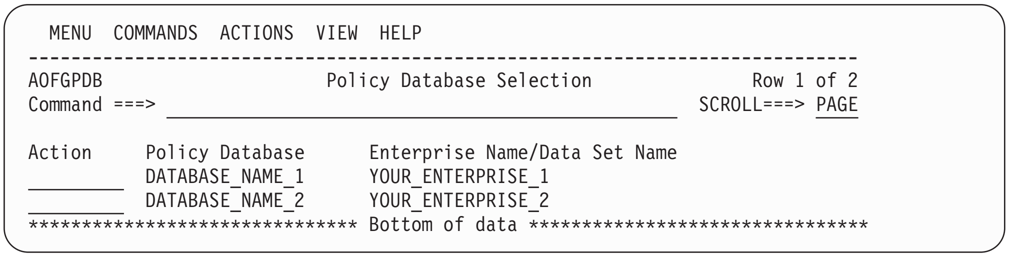
The I/O operations dialog panels are defined using Dialog Tag Language (DTL) for ISPF. Both the source panels and converted panels are provided in the product libraries. If you choose to update the panels, the source panels must then be reconverted.

## Step 13D: Verify the ISPF Dialog Installation

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | ✓ |

Logon to TSO using your modified logon procedure or running your data set allocation CLIST.

Access the customization dialog from the ISPF main menu that you defined. From the Customization Dialog Primary Menu that will appear, select option ***4 Policies*** to see a screen that looks similar to Figure 16.

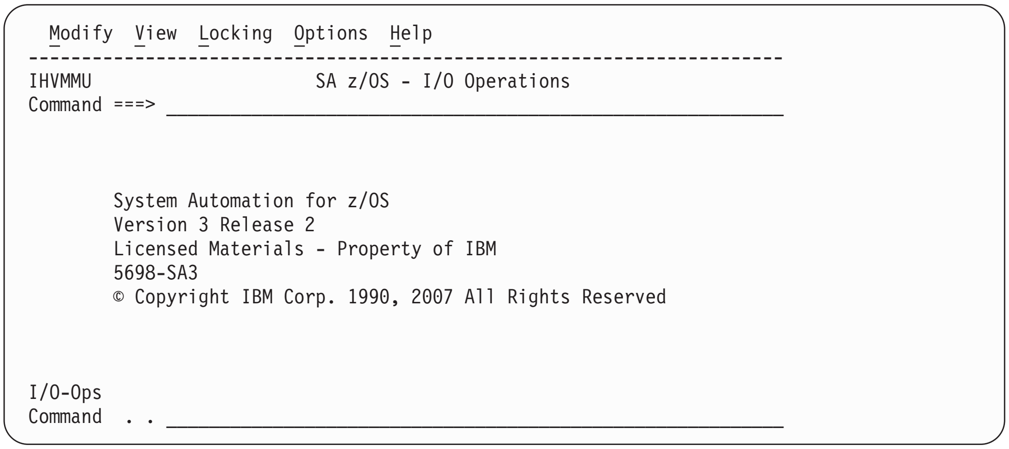


```
  MENU  COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
AOFGPDB                    Policy Database Selection                Row 1 of 2
Command ===> ________________________________________________ SCROLL===> PAGE

Action     Policy Database      Enterprise Name/Data Set Name
_________  DATABASE_NAME_1      YOUR_ENTERPRISE_1
_________  DATABASE_NAME_2      YOUR_ENTERPRISE_2
****************************** Bottom of data ********************************
```

*Figure 16. Policy Database Selection Screen*

A screen similar to the one shown in Figure 17 on page 119 will be displayed if you run the REXX exec IOCONNECT shown on page 118. You can use the information shown to verify your SA z/OS installation.



```
  Modify  View  Locking  Options  Help
----------------------------------------------------------------------
IHVMMU                    SA z/OS - I/O Operations
Command ===> __________________________________________________________



        System Automation for z/OS
        Version 3 Release 2
        Licensed Materials - Property of IBM
        5698-SA3
        © Copyright IBM Corp. 1990, 2007 All Rights Reserved



I/O-Ops
Command  . . __________________________________________________________
```

Figure 17. I/O Operations Initialization Panel

## Step 14: Verify the Number of available REXX Environments

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | |

Change the value of the maximum number of available REXX environments to at least 400. The variables to do this are in the sample assembly and linkedit job in SYS1.SAMPLIB(IRXTSMPE). Change the value of the ENTRYNUM= parameter to at least 400. The sample is a user exit, so follow your SMP/E process for handling user exits. See also “Allocation Requirements for REXX Environments” on page 32.

## Step 15: Customization of NetView for TEC Notification by SA z/OS

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| * | | |

This section describes the customization steps specific for TEC Notification by SA z/OS of all involved products:

- NetView
- SA z/OS

Depending on whether SA z/OS messages are forwarded to a local Message Adapter and Alert Adapter, or a message has to be forwarded to the SA z/OS focal point system running the Message Adapter, the NetView customization is different:

- In a *local configuration*, there is only one operator and you can use the default operator ID AUTOTEC.
- In a *distributed configuration*, you need to define a different operator ID on each target system. If the focal point is also configured as a target system that triggers messages and alerts, you need to define another different operator ID on the focal point itself. In case of a distributed configuration, you need to adapt the synonym table.

  All operator IDs of all target systems must be defined on the focal point.

Review the synonyms for TEC Notification by SA z/OS and set all listed synonyms to their appropriate value.

- %AOFTECTASK% and %AOFTECTASKQ%

  This is the name of the autotask for sending SA z/OS events to the Tivoli Enterprise Console. It is the operator ID you defined in your configuration. The default is `AUTOTEC`.
- %AOFTECPPI%

  This is the NetView PPI Receiver ID of the message adapter (with quotes). The default is `IHSATEC`.
- %AOFTECMODE%

  This is the event generation mode (with quotes). Possible values are:
  - LOCAL: the message adapter is running on *this* system. LOCAL is valid for the *local configuration* ("Environment Configurations" on page 45) and for the focal point in the *distributed configuration*.
  - REMOTE: the message adapter is running on a remote automation focal point. SA z/OS messages will be generated on **this** target system and forwarded to a *remote* automation focal point system. There is no local GEM message adapter which can process SA z/OS messages. REMOTE is valid for the target system in a *distributed configuration* ("Environment Configurations" on page 45).

  The default is `LOCAL`.

## Modifying Existing Files

Table 18 shows all product files which need to be modified.

*Table 18. Product Files to be Modified*

| File Name | DD Name | Description |
|---|---|---|
| AOFMSGSY | DSIPARM | Synonyms used in the AT |
| AOFOPFSO | DSIPARM | Operator definitions |

**AOFMSGSY**

Locate the automation fragment `AOFMSGSY` to update the required synonyms. See *IBM Tivoli System Automation for z/OS Customizing and Programming* for more information.

## Customizing the Automation Operators Policy Object

Define the automation operator AUTOTEC using the SA z/OS customization dialog **Automation Operator Definitions**.

For a complete description of the required dialogs, refer to *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

## Customizing the System Policy Object

You must define INGMTEC as an additional automation table, using the SA z/OS customization dialog System Information policy object (policy selection SYSTEM INFO).

For a complete description of the required dialogs, refer to *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

## Removing Messages

You may want to remove a message from the set of mapped messages. To do this, you only need to remove the *IF ... THEN* statement corresponding to the affected message from file *INGMTEC*.

## Customization of NetView Event/Automation Service

This section describes how to customize NetView Event/Automation Service, namely the message adapter and the alert adapter for messages and alerts from SA z/OS.

### Modifying Event/Automation Service Files

Table 19 shows all product files which need to be modified.

*Table 19. Product Files to be modified*

| Member Name | DD Name | Description |
|---|---|---|
| IHSAINIT | IHSSMP3 | initialization file for the Event/Automation Service |
| IHSAMCFG | IHSSMP3 | message adapter configuration file |
| IHSAACFG | IHSSMP3 | alert adapter configuration file |
| IHSAMFMT | IHSSMP3 | message adapter format file |
| IHSAACDS | IHSSMP3 | alert adapter CDS file |
| IHSAECFG | IHSSMP3 | Event Receiver Configuration file |

The following is a brief list of steps needed to customize the Event/Automation Service for SA z/OS specific message/alert routing. For detailed guidance see the chapter Customizing Event/Automation Service in the *Tivoli NetView Customization Guide* .

1. Adapt the initialization file for the Event/Automation Service IHSAINIT. At least the following values have to be defined:
   - *ALRTCFG*: specifies the alert adapter configuration, for example: `ALRTCFG=IHSAACFG`
   - *MSGCFG*: specifies the message adapter configuration file, for example: `MSGCFG=IHSAMCFG`
   - *PPI*: specifies the PPI receiver ID used by the Event/Automation Service, for example: `PPI=IHSATEC`
   - Make sure that the NOSTART statements for the tasks ALERTA and MESSAGEA are commented out using #.

   The values given here are examples only. They will be used throughout this chapter.

   **Note:** The *PPI* receiver ID for the Message Adapter specified here must be the same as the one defined in the synonym section of the NetView automation table.
2. Adapt the message adapter configuration file IHSAMCFG. You must at least define:

   ```
   ServerLocation=Hostname or IP address of your TEC Event Server
   ```

   If the port mapper is not available on the event server, the port number must be specified in the statement

   ```
   ServerPort=port number
   ```

AdapterFmtFile must be defined to INGMFMTE.

3. Adapt the alert adapter configuration file IHSAACFG. You must at least define:

   ```
   ServerLocation=Hostname or IP address of your TEC Event Server
   ```

   If the port mapper is not available on the event server, the port number must be specified in the statement

   ```
   ServerPort=port number
   ```

4. Insert the include statement shown in Figure 18 at the end of the Event/Automation Service format file IHSAMFMT. This will activate the message/event mapping defined in the message adapter format file *INGMFMT* for SA z/OS messages.

```
# ------------------------------------------------------------------ */
# System Automation for z/OS (AOF) message to TEC event mapping      */
# ------------------------------------------------------------------ */
%INCLUDE INGMFMT
```

*Figure 18. Format File Include Statement*

5. Insert the include statement shown in Figure 19 into the Event/Automation Service CDS file IHSAACDS to activate the alert / event mapping. Make sure the SA z/OS specific statements precede the more general statements of the same class. This can be achieved by inserting the include statement at the top of file *IHSAACDS*.

```
# ------------------------------------------------------------------ */
# System Automation for z/OS (AOF) message to TEC event mapping      */
# ------------------------------------------------------------------ */
%INCLUDE INGACDS
```

*Figure 19. CDS File Include Statement*

6. Adapt the Event Receiver Configuration file IHSAECFG. You must at least define: *NetViewAlertReceiver=NETVALRT*

## Step 16: Compile SA z/OS REXX Procedures

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| * | * | |

You should perform this step to gain considerable performance improvement for system operations startup.

You can optionally compile the SA z/OS automation procedures, which are written in REXX. The decision to compile the SA z/OS automation procedures implies an added responsibility for recompiling whenever ING.SINGNREX members are affected by SMP/E maintenance. To compile and execute these automation procedures, the IBM Compiler and Library for REXX/370 must be installed on your system along with their prerequisite products.

The JCL job INGEREXR and related routine INGEREXC are provided in the SA z/OS sample library to help you compile the ING.SINGNREX members. Modify the data set names and jobcard in INGEREXR as necessary and submit the job. The ING.SINGNREX.CREXX library can be modelled on ING.SINGNREX, and

ING.SINGNREX.LIST should be a VBA LRECL 125 PDS library. If necessary add to the SYSEXEC DD statement the library where the REXXC program can be found. Finally, specify the name of the resulting compiled-REXX data set in your NetView application startup procedure.

Consult the *REXX/370 User's Guide and Reference R3* (SH19-8160) for the compiler options that apply to your installation. If necessary, change the INGEREXC routine accordingly.

**Note:** SA z/OS has *not* been tested to run with the REXX Alternate Library. Officially, this is not a supported environment.

## Step 17: Defining Automation Policy

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | |

| Before you can start using automation, you need to define your automation policy
| using the customization dialog.

If you start from scratch, use the IBM samples delivered with the product and create your new policy database. Read the information in the section "Creating a New Policy Database" in *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

Next invoke the customization dialog to define your automation policy. Start by defining the following policy objects:

- Applications
- Application groups
- Processors
- Systems
- System defaults
- A group for each sysplex

You will find detailed information on how to perform these steps in *IBM Tivoli System Automation for z/OS Defining Automation Policy* which provides information on using the customization dialog for the required definitions.

If you already have a policy database, make a copy or backup, then complete the following steps.

### Step 17A: Build the Control Files

When the policies for the SA z/OS components have been defined, use the BUILD command to create the system operations control files (automation control file and automation manager configuration file, needed for automation), processor operations control file and NetView operator definitions. The BUILD command is available from various panels of the customization dialog. For more information on how to perform this step, refer to the manual *IBM Tivoli System Automation for z/OS Defining Automation Policy*. You can use the sample job INGEBBLD in the SINGSAMP sample library.

**Note:**
It is mandatory to use the SA z/OS customization dialog to create policy objects for the resources you want to automate. Do not edit the automation control files (ACF) manually.

A manually edited automation control file cannot be used to start SA z/OS.

## Step 17B: Distribute System Operations Control Files

The system operations control files consist of the automation control file and the automation manager configuration file. You need to make the control files available to the automation agents and automation managers on the target systems. All automation managers and automation agents in the same sysplex must have access to the same system operations control files or a copy of them. You must send the files to the target sysplexes and make the data available to the automation agents and the automation managers.

For the automation agents, it can either be in the DSIPARM concatenation or in a separate data set that has the same name as that known to the automation manager.

For the automation managers it can either be placed in the automation managers' current configuration data set or the automation managers can be told to use a new configuration data set.

# Step 18: Define Host-to-Host Communications

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | ✓ |

VTAM definitions are required for both host-to-host communications and host-to-workstation communications. This section of the installation addresses the host-to-host communications.

Verify that your NetView APPL member is consistent with the steps that follow.

The host-to-host communications require:
- Defining each host as a CDRM
- Defining the host ACB

## Step 18A: Customize the SYS1.VTAMLST Data Set

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | |

Edit the member that defines NetView to VTAM and do the following:
1. Include as many NetView operator subtask APPL statements as operators that you defined in the DSIOPF member of the NetView DSIPARM data set.

2. SA z/OS uses the NetView BGNSESS command with the parameter SRCLU=* to create terminal access facility (TAF) fullscreen sessions for communication with OMEGAMON monitors, if requested.

   **Note:** It is expected that OMEGAMON classic is installed and has been configured for VTAM.

   Include one model terminal access facility (TAF) APPL statement to let NetView define the application dynamically, for example:

   ```
   TFxx#*    APPL MODETAB=AMODETAB,EAS=9,                              X
                  DLOGMOD=M2SDLCNQ
   ```

   where xx are the last two characters of the domain ID. See *Tivoli NetView for z/OS Installation: Configuring Additional Components* and *z/OS Communications Server: SNA Network Implementation Guide* for more details.
3. Define the NetView primary program operator interface task (PPT) as AUTH=(NVPACE,SPO). This causes unsolicited VTAM messages to be broadcast on the SSI and thus to be available to NetView.

   If, however, you have another NetView defined as a primary program operator application program (PPO), it receives unsolicited messages first and messages do not reach the NetView that is defined as a secondary program operator application program (SPO). See *Tivoli NetView for z/OS Installation and Administration* for information on PPO and SPO definitions.

For each target hardware that is defined with an SNA-based NVC connection to the processor operations focal point, VTAM majornode definitions are required to enable the hardware access for processor operations. Appendix D, “Processor Operations Sample,” on page 199 illustrates this for an OSA adapter that is the SNA gateway for the Support Elements, and the definition of an SE as a VTAM Switched Majnode. For other VTAM definition examples, refer to *Managing Your Processors*, (GC38-0452-08).

## Step 18B: Perform VTAM Definitions

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| | | ✓ |

**Note:** This applies to I/O operations host-to-host communications only. If you have configured a prior level of ESCON Manager or I/O operations, these definitions remain the same.

To use VTAM for I/O operations, there are some definitions that VTAM requires. These definitions are in addition to those needed for the installation and running of VTAM. If you already have VTAM installed, some of these definitions may already exist.

The I/O operations program in each host that carries on this communication must be defined as a VTAM application in each host. The I/O operations program that it communicates with in another host must be defined as a cross domain resource unless you use APPN. I/O operations uses the LU 0 protocol for the communication between hosts.

Because the means of the I/O operations program may be a channel-to-channel adapter, this connection has to be defined to VTAM via VTAM definition statements.

If the alternate path used is via a network communications program (NCP), the NCP must be defined to VTAM.

In order for VTAM to choose what routes to use for this communication and what priorities to assign, PATH statements and CLASS OF SERVICE must be defined.

An example of some of these VTAM definition statements is shown in Figure 20.

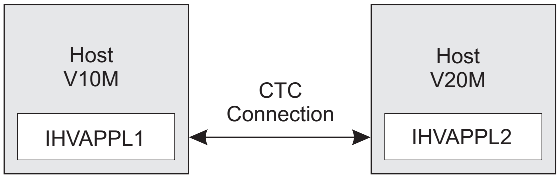


*Figure 20. VTAM Definition Statements*

In this example, there are two hosts running I/O operations. One application is named IHVAPPL1 and is in subarea 10. The second application is named IHVAPPL2 and is in subarea 20. Each host has its own set of VTAM definition statements.

## Cross-domain definitions

```
V10M                                   V20M

VTAMA    VBUILD TYPE=CDRM              VTAMB    VBUILD TYPE=CDRM
V10M     CDRM   SUBAREA=10             V10M     CDRM   SUBAREA=10
V20M     CDRM   SUBAREA=20             V20M     CDRM   SUBAREA=20
```

The appropriate definitions are needed for each host that will be communicating via I/O operations. Each host will be defined as a CDRM.

If a communication path between the hosts is a channel-to-channel adapter, this has to be defined to VTAM.

**Note:** Change each “*x*” to the appropriate value.

```
CTCV20 VBUILD TYPE=CA
label1 GROUP LNCTL=CTCA,
             DELAY=x,
             MIH=x,     (cause link to INOP if SIO timeout occurs)
             REPLYTO=x  (tells VTAM how long to wait for completion after
                         channel program started)
label2 LINE  ADDRESS=x, (channel unit address of channel to channel adapter)
             MAXBFRU=x  (# of buffers VTAM will use to receive data)
label3 PU    PUTYPE=4,
             TG=1
```

Each I/O operations program must be defined via an application statement in each host. The user-specified names must be unique in the network. These are the names that the other I/O operations hosts will know each I/O operations by.

The ACBNAME parameter is required for I/O operations. This name must be IHVISC, and must be reserved for this use only.

The parameters SONSCIP=YES and AUTH=ACQ must also be specified.

For I/O operations it is strongly recommended that the DLOGMOD and MODETAB parameters given in the example below, or equivalent definitions, should be used. Note that an RUSIZE of 'zero' is used with this LU TYPE 0 protocol.

```
         VBUILD TYPE=APPL                        VBUILD TYPE=APPL
IHVAPPL1 APPL  ACBNAME=IHVISC,        IHVAPPL2 APPL  ACBNAME=IHVISC,
               AUTH=ACQ,                             AUTH=ACQ,
               DLOGMOD=INTERACT,                     DLOGMOD=INTERACT,
               SONSCIP=YES,                          SONSCIP=YES,
               MODETAB=ISTINCLM                      MODETAB=ISTINCLM
```

Using the above VTAM definitions the LOGMODE table entry would be:

```
                IBM3767 MODEENT LOGMODE=INTERACT,
                              FMPROF=X'03',
                              TSPROF=X'03',
                              PRIPROT=X'B1',
                              SECPROT=X'A0',
                              COMPROT=X'3040'
```

Each host must have a cross-domain definition for the other I/O operations host applications. They are defined as cross domain resources, as follows:

```
         VBUILD TYPE=CDRSC                      VBUILD TYPE=CDRSC
IHVAPPL2 CDRSC CDRM=V20M               IHVAPPL1 CDRSC CDRM=V10M
```

The communication paths between the I/O operations hosts must be defined, as follows:

```
         PATH DESTSA=20,                        PATH DESTSA=10,
              ER0=(20,1),                            ER0=(10,1),
              ER1=(20,1),                            ER1=(10,1),
              VR0=1,                                 VR0=1,
              VR1=0                                  VR1=0
```

The class of service (COS) definition is:

```
ISTSDCOS COSTAB                        ISTSDCOS COSTAB
         :                                      :
IHVAPPL1 COS VR=((0,2),(1,2))          IHVAPPL2 COS VR=((0,2),(1,2))
         :                                      :
         COSEND                                 COSEND
```

## APPN Definitions

Assuming that both hosts reside in the same domain, XYZ, the equivalent APPN definitions are:

```
 VTAM A                              VTAM B

                   APPN Transport Resource List

         VBUILD TYPE=LOCAL                    VBUILD TYPE=LOCAL
XYZANTRL PU TRLE=XYZATRLN,           XYZBNTRL PU TRLE=XYZBTRLN,
            CONNTYPE=APPN,                       CONNTYPE=APPN,
            CPCP=YES                             CPCP=YES

                   Channel to Channel Adapter

         VBUILD TYPE=TRL                      VBUILD TYPE=TRL
XYZATRLN TRLE LNCTL=MPC,             XYZBTRLN TRLE LNCTL=MPC,
              READ=(cua),                          READ=(cua),
              WRITE=(cua),                         WRITE=(cua),
              MPCLEVEL=NOHPDT                      MPCLEVEL=NOHPDT

                   I/O operations / VTAM
 __________________________________________________________________________
          VBUILD TYPE=APPL             |          VBUILD TYPE=APPL
 IHVAPPL1 APPL ACBNAME=IHVISC,         | IHVAPPL2 APPL ACBNAME=IHVISC,
               AUTH=ACQ,               |               AUTH=ACQ,
               DLOGMOD=INTERACT,       |               DLOGMOD=INTERACT,
               SONSCIP=YES,            |               SONSCIP=YES,
               MODETAB=ISTINCLM        |               MODETAB=ISTINCLM
```

For details of the channel unit address (CUA) refer to the section “Operand descriptions” in the chapter “Transport resource list major node” in *z/OS Communications Server: SNA Resource Definition Reference*.

The APPL-specific definitions are identical to those described in “Cross-domain definitions” on page 126.

## Step 18C: Perform TCP/IP Definitions

| SysOps | ProcOps | I/O Ops |
|---|---|---|
|  |  | ✓ |

**Note:** This applies to I/O operations host-to-host communications only. These definitions are new.

In order to use TCP/IP for I/O operations, there are some definitions that I/O operations requires. These definitions are in addition to those needed for the installation and running of VTAM.

With SA z/OS 3.2 I/O operations prefers to communicate with other hosts using the TCP protocol when the remote host is running a release level that is the same or higher. I/O operations requires a TCP/IP host name or an alias name of not more than 8 characters. This is because this name must be stored in the switch host data buffer, which has a limited size. If you have defined longer host names you must define an alias of up to 8 characters for each host name that exceeds the limit.

1. Define each host alias. Refer to the section “Host alias table” in “Chapter 2. Configuration overview” of *z/OS Communications Server: IP Configuration Guide* for the description of how to define the table entries such as:

   ```
   IHVSYS1 BOEBSYST1.BOEBLINGEN.DE.IBM.COM
   IHVSYS2 BOEBSYST2.BOEBLINGEN.DE.IBM.COM
   ```

2. Check the LOOKUP statement of each system that you have defined an alias for. Verify that either LOCAL is specified, or LOCAL precedes DNS because the DNS server does not return an alias name. Refer to the section “LOOKUP statement” in “Chapter 5. TCPIP.DATA configuration statements” of *z/OS Communications Server: IP Configuration Reference* for more information.
3. Update the startup procedure of the Resolver. Refer to “Resolver customization” in “Chapter 2. Configuration overview” of *z/OS Communications Server: IP Configuration Guide* for a sample procedure:

   ```
   :
   //* Function: Start Resolver
   //*
   //EZBREINI EXEC PGM=EZBREINI,REGION=0M,TIME=1440,PARM=(&PARMS,
   //      'ENVAR("HOSTALIASES=//''data_set_name''")/-d 0')
   //*
   //* When the Resolver is started by UNIX System Services it is
   ```

```
//* started with SUB=MSTR.
   ⋮
```

4. Restart the TCP/IP address space and the Resolver address space.

Because each I/O operations program acts as a server as well as a client, it requires port definitions in the *hlq*.ETC.SERVICES data set:

```
IHVsrvr  portnumber/TCP
IHVclnt  portnumber/TCP
```

The first entry is mandatory for reestablishing a connection after an interrupt. The second entry is optional. You have to specify the second entry if you want to restrict particular ports for the use by I/O operations. For details of controlling access to ports refer to the section "Port access control" in "Chapter 3. Security" of *z/OS Communications Server: IP Configuration Guide*.

Note that the service names require the component code to be defined in upper case and the remaining characters in lower case. For details refer to *z/OS Communications Server: IP Configuration Reference*.

**Note:** If you omit the definition of the server port, I/O operations suppresses the TCP/IP communication for its lifetime and falls back to VTAM communication.

Check the MAXSOCKETS parameter in the BPXPRM*xx* parmlib member's NETWORK statement that corresponds to the addressing family. This value determines how many sockets for a particular addressing family can be opened in the entire system. I/O operations requires twice the number of possible TCP connections plus one.

The number of sockets than an application can open is also limited by the UNIX System Services parameter MAXFILEPROC in the BPXPRM*xx* parmlib member. This parameter determines the number of sockets each address space can have open. The same rules apply to this parameter, that is, I/O operations requires twice the number of possible TCP connections plus one.

Check the SOMAXCONN value that defines the maximum number of connection requests queued for the listening socket. I/O operations cannot have more than 256 connection (VTAM and TCP/IP) at a time. However it is very unlikely that all connection requests occur at the same time because it is dependent on the time that each I/O operations is started. The default value of 10 is probably large enough.

## Step 19: Enabling SA z/OS to Restart Automatic Restart Manager Enabled Subsystems

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | | |

If you intend to use the z/OS Automatic Restart Manager and you want to coordinate its actions with those of SA z/OS, you must ensure the following:

- The SA z/OS-supplied element restart exit (ERE) must be available to z/OS. The exit, AOFPERRE, is in the ING.SINGMOD2 data set. No customization is required.

- The AOFARCAT autotask must be created. The autotask name is included in the AOFOPF member and is created automatically by NetView if you install SA z/OS without changing AOFOPF.
- The NetView Subsystem Interface (SSI) must be active for the coordination of SA z/OS and z/OS automatic restart management to occur.
- As part of its Automatic Restart Manager support, SA z/OS claims all PPI receiver IDs starting with AOF. If you have any other PPI receivers named AOF*xxxx*, results are unpredictable.

For further information on the relationship between SA z/OS and Automatic Restart Manager, see *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

## Step 20: Define Security

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | |

**Note:**
To plan your RMTCMD-based INGSEND security, see the discussion of RMTCMD security features in the NetView library.

You should perform this step if you want to ensure that only authorized staff can manage the resources in your environment.

Your operations staff and automation facilities at SA z/OS-controlled systems need to be authorized to manage the resources in their environment. You can control human and automation operator authority through the password security provided by either by NetView or an SAF-based security product, such as RACF.

Additionally SA z/OS provides the following samples in the SA z/OS SINGSAMP sample library to help you to establish security on your systems:

1. INGESAF sample JCL with definitions for a RACF environment
2. INGESCAT sample command authorization table for security checking within NetView

See also the following sections in Appendix A, “Security and Authorization,” on page 171 for other security options:

- “Granting NetView and the STC-User Access to Data Sets” on page 171
- “Restricting Access to INGPLEX and INGCF Functions” on page 173
- “Security for IBM Tivoli Monitoring Products” on page 174 (OMEGAMON)
- “Controlling Access to the Processor Hardware Functions” on page 178
- “Defining an RACF Profile for I/O Operations” on page 180
- “Establishing Authorization with Network Security Program” on page 182

For SNMP, BCP internal interface, and TCP/IP connections, it is mandatory to make the security definitions described in “Controlling Access to the Processor Hardware Functions” on page 178.

For UNIX System Services automation, one or more UNIX segments (OMVS) must be defined. For details, refer to “Step 31A: Define UNIX Segments (OMVS)” on page 151.

## Step 21: Customize the Status Display Facility (SDF)

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| * | | |

If you decide to use SDF as the SA z/OS fullscreen operator interface for monitoring automated resource statuses at the NetView 3270 console, customizing SDF involves defining the following:

- SDF initialization parameters. These are defined in the AOFINIT member of a NetView DSIPARM data set.
- Resource hierarchy or tree structure. The AOFTREE member of a NetView DSIPARM data set includes the appropriate tree members, which contain the resource hierarchy information.
- Color and priority assignments for resource status types. These have default values that are set up by SA z/OS (see *IBM Tivoli System Automation for z/OS User's Guide* for details), but you can define overrides to color and priority assignments with the SA z/OS customization dialog.
- SDFROOT. You can specify a root name for the SDF tree on the Environment Setup Panel of the customization dialog. If you do not specify a new root name, it defaults to the value specified for SYSNAME.

See *IBM Tivoli System Automation for z/OS Customizing and Programming* for detailed information about customizing SDF.

## Step 22: Check for Required IPL

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | ✓ |

An IPL is only required if:

- In “Step 4D: Update LPALST*xx*” on page 82 you decided ***not*** to use the solution to dynamically add the modules to the LPALST
- In “Step 4E: Update LNKLST*xx*” on page 83 you updated LNKLST and you decided ***not*** to use the solution to dynamically add the modules to the LNKLST
- “Step 4F: Update IEFSSN*xx*” on page 83 was required because the IEFSSN*xx* member was not updated during NetView installation and you cannot use the z/OS command SETSSI for a dynamic update of the subsystem name table.

## Step 23: Automate System Operations Startup

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| ✓ | ✓ | |

Add commands to the COMMND*xx* member of SYS1.PARMLIB to start the automation NetView when z/OS starts. You may also need to modify an IEASYS*xx* member of SYS1.PARMLIB to specify which COMMND*xx* or other PARMLIB members to use during IPL. SA z/OS initialization begins with starting system operations. If an SA z/OS automation policy is used, system operations subsequently starts processor operations and I/O operations.

Make the described changes to the following SYS1.PARMLIB data set members:

**COMMND*xx***

Make sure that the procedure names you choose match those specified in the SYS1.PROCLIB data set.

Compare the contents of the COMMND*xx* member with the INGECOM member which resides in the SINGSAMP sample library. Edit the COMMND*xx* member and do the following:

1. If you want to use the recording of IPL function (INGPLEX IPL command) add the following statement in the COMMND*xx* member:

   ```
   COM='S HSAPIPLC,SUB=MSTR'
   ```

   This procedure collects the IPL information in MVS. Return codes for this procedure are documented in the HSAPIPLC sample.
2. If you are running more than one NetView on your system, ensure that you have included start commands for the Automation NetView.

   ```
   COM='S INGENVSSI,SUB=MSTR'
   COM='S INGENVSA,SUB=MSTR'
   ```

   **Note:**

   NETVSSI here is a placeholder for the name of the member to which you copied the NetView subsystem interface startup procedure in "Step 6: Customize SYS1.PROCLIB Members" on page 87.

   NETVSTRT here is a placeholder for the name of the member to which you copied the NetView application startup procedure in "Step 6: Customize SYS1.PROCLIB Members" on page 87.

This adds commands that select the correct MPF entries and that start NetView.

**IEASYS*xx***

Edit the IEASYS*xx* member to specify which SYS1.PARMLIB data set members to use during the IPL process. This is done by specifying the 2-character suffix of the SYS1.PARMLIB member names. If you choose SO, the statements in the IEASYS*xx* member would be as follows:

```
APF=SO
CMD=SO
CON=SO
SSN=SO
SCH=SO
LNK=SO
LPA=SO
```

For example, because APF=SO, the system uses the IEAAPFSO member during the IPL process.

## How to Automate the Automation Manager Startup

**Note:** The system on which the automation manager should be started must be defined as policy object System in the policy database which will be used to create the automation manager configuration file that this automation manager uses (see also “Step 17A: Build the Control Files” on page 123.

To enable automatic startup of the automation manager whenever SA z/OS is started, add the following start command for the automation manager to the COMMND*xx* PARMLIB member, where *procname* is your selected name of the automation manager start procedure:

```
S procname,SUB=MSTR
```

You can find a sample startup procedure called INGEAMSA in SINGSAMP. sample library, so that your entry in the COMMND*xx* member could look as follows:

**Sample COMMNDxx entry**

```
'S INGEAMSA,JOBNAME=HSAM&SYSCLONE.,SUB=MSTR'
```

## How to Automate WebSphere MQ Startup

**Note:** This substep is not necessary when you have decided to use XCF for communication between the automation manager and the automation agents.

When you use WebSphere MQ for manager-agent communication and status backup, you can automate WebSphere MQ and let it be started and stopped by SA z/OS (for details on how this is made possible, see “Setting up WebSphere MQ V5.3 (Optional)” on page 37).

In a full sysplex environment it is recommended that you start both the local WebSphere MQ manager and its associated DB2 together immediately after JES is up and running. In a single system case with WebSphere MQ version 2.1, DB2 is not needed. See the related product installation manuals for information on how to start WebSphere MQ and DB2 and define these resources to SA z/OS in the customization dialog. For more information, also refer to “Peer Recovery Considerations” on page 41.

Consider that the subsystem RRS is also necessary for shared DB2 database functions.

# Step 24: Verify Automatic System Operations Startup

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| * | | |

After you have installed the host components of SA z/OS, it is recommended that you perform the following steps for verification purposes:

1. Perform an IPL, if you have not done this according to “Step 22: Check for Required IPL” on page 131. Then start SA z/OS and perform a coldstart. A coldstart is performed by default unless you specify the warmstart option. Do

not select the warmstart option, because you might have policy data from earlier releases in your warmstart cache.

The following messages should appear on the system console:

```
AOF532I hh:mm:ss AUTOMATION ENVIRONMENT HAS BEEN INITIALIZED
AOF540I hh:mm:ss INITIALIZATION RELATED PROCESSING HAS BEEN COMPLETED
```

2. Use the NetView LIST command to confirm that the following SA z/OS tasks are active:

| Task Name | Description |
|---|---|
| AOFTSTS | automation status file task |
| INGPXDST | XCF communication task |

To confirm that these tasks are active, log on to NetView, and enter the NetView LIST command to display the status for each task:

```
LIST taskname
```

3. Use the commands INGAMS and INGLIST to verify that they work.
4. Check that the subsystem status and automation flag settings are what you expect. Enter the DISPSTAT ALL command to display the status of automated subsystems and the DISPFLGS command to display the automation flag settings: See *IBM Tivoli System Automation for z/OS Operator's Commands* for information about these commands.
5. Use the SA z/OS DISPAUTO command in NetView to display a menu that allows you to initiate further command dialogs. These display information about your automation. Enter DISPAUTO and then choose one of the menu options. See *IBM Tivoli System Automation for z/OS Operator's Commands* for information about the DISPAUTO command.
6. Confirm that the automation shuts down and restarts the subsystems as you expect. You can shutdown and restart each automated resource individually using the following SA z/OS command:

```
INGREQ resource REQ=STOP SCOPE=ONLY RESTART=YES
```

If any of the resources (subsystems) do not restart as you expect, make corrections to your automation policy.

## Step 25: Install an SA z/OS Satellite

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| * | | |

This step is only required if your enterprise runs an Automation NetView and a Networking NetView with GMFHS on the focal point system or on another focal point NetView. You must then install SA z/OS on the automation NetView that is used for system automation.

### Step 25A: Customize the Networking NetView or Focal Point NetView Startup Procedure

In SYS1.PROCLIB or another procedure library, find members used to start the Networking NetView application. Insert the data set names from the following table into the indicated DD concatenations.

**Notes:**

1. The data sets listed in Table 20 should appear last in your concatenation. If they appear before other data sets (for example, data sets containing members customized for automated network operations [AON/MVS]), results are unpredictable.
2. The ING.SINGMOD1 library needs to be authorized for *APF*.

*Table 20. Members to Start the Networking NetView*

| DDNAME | System Operations Data Set |
|---|---|
| STEPLIB | ING.SINGMOD1 |
| DSICLD | ING.SINGNREX |
| DSIPARM | ING.SINGNPRM |
| DSIMSG | ING.SINGNMSG |
| DSIPRF | ING.SINGNPRF |
| CNMPNL1 | ING.SINGNPNL |

## Step 25B: Customize the Networking NetView or Focal Point NetView DSIPARM Data Set

Several members in the DSIPARM concatenation must be customized for the SA z/OS satellite. Before editing an SA z/OS member, remember to copy it from ING.SINGNPRM into a new, user-defined data set that is placed before ING.SINGNPRM in the concatenation.

**NetView style sheet**

To enable SA z/OS, make sure that the following TOWER statements are activated in the NetView style sheet:

- For a satellite SA z/OS on a Networking NetView:

```
TOWER = SA
TOWER.SA = SATELLITE
```

- For full SA z/OS:

```
TOWER = SA
TOWER.SA = SYSOPS
```

**AOFMSGST**

If you do not choose to use the NetView operator IDs defined by SA z/OS, copy and edit AOFMSGST to contain the appropriate definitions of the synonyms %AOFOPMSU%, %AOFOPHB% for your Networking NetView. %AOFOPMSU% is a synonym for the operators that can be routed commands as a result of alerts trapped in the NetView automation table. %AOFOPHB% is a synonym for the operator that can be routed heartbeat alerts trapped in the NetView automation table. (Note that there can be only one operator defined for %AOFOPHB% and it must be unique and not used for any other functions). Other synonyms in the member are not specific to the Networking NetView environment.

**AOFRODM**

Copy and edit AOFRODM to contain the correct name for your RODM and a user ID authorized to update it.

- Specify a RODM name by changing RODMNAME=NONE to RODMNAME=xxxxxxxx, where *xxxxxxxx* is your RODM name.

- Specify a user ID by changing RODMUSER=XXAOCFR to RODMUSER=xxxxxxxx, where *xxxxxxxx* is your user ID for batch updates from NetView.

## Step 26: Installing and Customizing the NMC Focal Point

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| * | * | |

Communication between SA z/OS and the NMC focal point is maintained by the SA z/OS topology manager. An SA z/OS topology agent on each target system retrieves the enterprise data from the automation manager. The SA z/OS topology manager on the focal point provides the information into RODM. GMFHS takes the information from RODM and presents it in a graphical form on the NMC workstation. This information is available if you have completed the previous installation steps, that is, SA z/OS is fully functional.

There are two possible configurations when setting up the NMC focal point:

1. Full SA z/OS
2. A satellite SA z/OS on a Networking NetView

The following sections describe how to customize the SA z/OS topology manager for the operators.

### Step 26A: Preparing for NMC

Some of the tasks in this step are different for full SA z/OS and a satellite SA z/OS, as indicated.

1. **Applications Required by NMC**

   The applications that NMC uses must be available, so make sure of the following:

   - RODM is running and the RODM load function has loaded the data model into RODM
   - GMFHS and MultiSystem Manager are installed and working

   For information on how to do this, refer to

   - *Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*
   - *Tivoli NetView for z/OS Graphic Monitor Facility User's Guide*
   - *Tivoli NetView for z/OS MultiSystem Manager User's Guide*

2. **Configuration**

   Perform the following configuration:

   - **Full SA z/OS:** Import the *NMC sample add-on policy that is delivered with SA z/OS into your policy database and customize its definitions there to fit your environment.
   - **SA z/OS Satellite:** You must start the environment manually.

   SA z/OS delivers a NetView automation table fragment AOFMSGST that automates this setup.

   - **Full SA z/OS:** You must define this fragment in the customization dialog to be loaded on the focal point only. With this table, the SA z/OS topology manager is started after the completion message from MultiSystem Manager.

Alternatively, you can specify the following statement in the NetView style sheet:

```
TOWER.SA = SYSOPS SATELLITE
```

- **SA z/OS Satellite:** Specify the following statement in the NetView style sheet:

  ```
  TOWER.SA = SATELLITE
  ```

For additional information, refer to the description of the NetView style sheet and AOFMSGST in "Step 25B: Customize the Networking NetView or Focal Point NetView DSIPARM Data Set" on page 135.

3. **Security**

   For security considerations, refer to "Securing Focal Point Systems and Target Systems" on page 171.

4. **NetView Operator Tasks**

   The RODM name and RODM user must be customized in member AOFRODM on the focal point system (see "Step 25B: Customize the Networking NetView or Focal Point NetView DSIPARM Data Set" on page 135). Customizing AOFRODM on any other system is not necessary.

   - **Full SA z/OS:** You must define the following Automation Operators in the customization dialog for *both* the satellite and target systems:
     - EVTOPER (the default for the primary is AUTEVT1 and for the backup is AUTEVT2)
     - HBOPER (the default is AUTHB)
     - HBSLV (the default is AUTBSLV)
     - POSTOPER (the default is AUTPOST)
     - POSTSLV (the default is AUTPOSTS)

     **Note:** If the task defined on the target systems is different to the task defined on the satellite, INGSEND definitions must be defined with the customization dialogs to provide the mapping by using the SEND COMMAND OPERS policy item of the Enterprise policy object.

     These automation operator definitions are included in the *NMC sample add-on policy. Customize them there to fit your environment.

     You must also define these automation operators in DSIOPF or RACF or both.
   - **SA z/OS Satellite:** No customization is required because this is done automatically by SA z/OS during the initialization of the satellite.

5. **SA z/OS Data Storage**

   Two repositories are provided for SA z/OS data:

   - The automation manager (for target systems)
   - RODM (for the focal point)

6. **NetView Common Global Variables**

   You must set the following NetView common global variables for the *target* system. Set them in SINGNPRM(AOFSTYLE). The defaults are underlined.

   **AOFUPDAM**

   Determines whether SA z/OS data should be stored in the automation manager:

| | | Can Be Set For: | |
|---|---|---|---|
| Value | Meaning | SA z/OS | Satellite |
| YES | SA z/OS data is stored in the automation manager. | ✓ | X |
| **NO** | SA z/OS data is *not* stored in the automation manager. | ✓ | ✓ |

**AOFUPDRODM**

Determines whether SA z/OS data should be stored in RODM:

| | | Can Be Set For: | |
|---|---|---|---|
| Value | Meaning | SA z/OS | Satellite |
| **YES** (NMC user) | SA z/OS data is stored in RODM. | ✓ | ✓ |
| NO (non-NMC user) | SA z/OS data is *not* stored in RODM. | ✓ | ✓ |

**AOFSENDALERT**

Defines the mechanism that is used to forward data from the target to the focal point. It is only relevant if AOFUPDRODM has been set to YES.

| | | Can Be Set For: | |
|---|---|---|---|
| Value | Mechanism | SA z/OS | Satellite |
| YES | Alerts | ✓ | ✓ |
| **NO** | Command Handler | ✓ | X |

Setting the values of AOFUPDAM, AOFUPDRODM and AOFSENDALERT on a Networking NetView (for a satellite) or Focal Point NetView is not necessary because this is done automatically.

AOFUPDAM, AOFUPDRODM and AOFSENDALERT must be set to the same value on each target system in a sysplex.

AOFUPDAM used in conjunction with AOFUPDRODM will control if and where the SA z/OS data is stored, as shown in Table 21.

*Table 21. Use of AOFUPDAM and AOFUPDRODM to Control SA z/OS Data Storage*

| AOFUPDAM | AOFUPDRODM | Storage Outcome | Usage |
|---|---|---|---|
| YES | YES | SA z/OS data stored in the automation manager and also in RODM. | NMC user, any loss of contact between the target systems and the focal point will be followed by the RODM data being rebuilt from the SA z/OS data that had previously been stored in the automation manager, this will ensure no loss of SA z/OS data shown on the NMC. |

*Table 21. Use of AOFUPDAM and AOFUPDRODM to Control SA z/OS Data Storage (continued)*

| AOFUPDAM | AOFUPDRODM | Storage Outcome | Usage |
|---|---|---|---|
| YES | NO | SA z/OS data stored in the automation manager only. | Non-NMC user, it is possible to create a feed from the SA z/OS data held in the automation manager (not used at present, may be used in future releases of SA z/OS). |
| NO | YES | SA z/OS data stored in the RODM only. | NMC user, no requirement to rebuild the RODM SA z/OS data. |
| NO | NO | SA z/OS data not stored in the automation manager or in RODM. | Non-NMC user. |

## Step 26B: Modify the NetView DSIPARM Data Set for the SA z/OS Topology Manager

There are a few things you have to do to prepare for the SA z/OS topology manager to run. Table 22 lists the data sets to be modified for this purpose.

*Table 22. DSIPARM Members to be modified for the SA z/OS Topology Manager*

| DSIPARM Member | Description |
|---|---|
| AOFOPFFP | System operations automation operator definitions |
| CNMSTYLE/C*xx*STGEN | NetView system level parameters for NetView initialization |
| DSI6INIT | Initialization member for the NetView DSI6DST task. |
| DSICRTTD | NetView CNM router initialization member |
| DUIFPMEM | NetView focal point definitions |
| DUIGINIT | GMFHS initialization member |
| FLCSAINP | MultiSystem Manager initialization member |
| INGTOPOF | NMC definition member |

### CNMSTYLE/CxxSTGEN

**Note:** This is only necessary if you have chosen to use alert forwarding as your communication method.

To avoid further changes, alert forwarding `ALERTFWD NV-UNIQ` is recommended. However, any of the following SNA-MDS settings can be defined:

- ALERTFWD SNA-MDS=LOGONLY
- ALERTFWD SNA-MDS=AUTHRCV
- ALERTFWD SNA-MDS=SUPPRESS

Although SNA-MDS is not absolutely required, it might be important as it allows the construction of networks with intermediate focal points and hot backups.

If the network contains an intermediate focal point, ALERTFWD SNA-MDS must be specified in CNMSTYLE/C*xx*STGEN. If the network does not contain an intermediate focal point, ALERTFWD NV-UNIQ may be specified in CNMSTYLE/C*xx*STGEN.

If ALERTFWD SNA-MDS is specified in CNMSTYLE/C*xx*STGEN, the following entries must be added to sample BNJRESTY:

```
E0 AUTO  SYSTEM AUTOMATION FOR z/OS
E1 DOMN  SYSTEM AUTOMATION FOR z/OS
E2 NET   SYSTEM AUTOMATION FOR z/OS
```

**Note:** The three values shown above (’E0’,’E1’, and ’E2’) are the first three user-defined values. If you already have user-defined entries in BNJRESTY, you may use alternative values for these entries.

For more information on how to add user-defined entries (E0 - EF) to BNJRESTY, refer to the following chapters in *Tivoli NetView for z/OS Customization Guide*:

- Customizing Hardware Monitor Displayed Data
- Using NMVT Support for User-Written Programming
- Adding or Modifying Resource Types

For more information about the ALERTFWD statement, refer to *Tivoli NetView for z/OS Administration Reference*.

## DSI6INIT

This is the initialization member for the NetView DSI6DST task and needs to have the appropriate focal point defined.

```
     DEFFOCPT TYPE=ALERT,PRIMARY=NETA.CNM02,BACKUP=NETA.CNM03
```

Note that on the focal point and the backup you will need different members, as NetView complains if a definition references its own system.

Usage of the LU 6.2 alert forwarding mechanism allows for the construction of focal point networks that include intermediate focal points.

## Autotask Operator IDs

Each focal point that will be running the SA z/OS topology manager must have an autotask defined for it. Your environment may have one or more of the following types of focal point:

- The primary focal point
- The secondary focal point
- The intermediate focal point (IFP)

This requires a definition in DSIPARM.DSIOPF, as follows:

```
  &domain.TPO   OPERATOR    PASSWORD=&domain.TPO
                PROFILEN    AOFPRFAO
```

This definition must be made on the focal point (or focal points) and on each target system. It should only be started as an autotask on the focal point.

An include member, DSIPARM.AOFOPFFP, has been provided to help you centralize and manage these operator IDs. You need to customize it to contain the operator IDs for your focal points.

The &*domain*. variable contains the focal point's domain ID. This is just a suggestion for the naming scheme.

**Note:** The names must be unique on the focal point and the target systems.

Additionally, on the focal point, the operator ID must be defined in the DSIPARM.AOFMSGST member, as the value for the %AOFOPTOPOMGR% synonym.

```
   SYN %AOFOPTOPOMGR%  = '&domain.TPO';
```

You should not include any backup operators in this synonym.

Installing and customizing needs to be done on the NMC focal point system or on each target system. (This is only for ProcOps.)

It is recommended to use system symbols for the focal point, backup, and intermediate focal point specification. In this case, you can update AOFOPFFP and AOFMSGSY accordingly and make it available in a general data set to all your systems, focal points, and targets. This avoids the same specification of two members on any single system.

You will need one set of autotasks for your primary focal point and a second set for your backup focal point. If you are using intermediate focal points, you will also need a set of operators for each of those (but only on the target systems that are defined to the IFP). Note that even in an IFP situation, the focal point will contact all target systems directly to obtain status and configuration data. The IFP is only used for alert forwarding.

## Operator Profiles

This concerns statements in the NetView operator definition file (DSIOPF), which associate operator IDs with logon profiles and the profiles themselves, which are defined in the DSIPRF concatenation.

Each operator who will be an NMC Administrator must be assigned a NetView logon profile which includes the NGMFADMN=YES key/value pair on its AUTH tag.

Each NMC user who needs to issue commands against resources through the NMC interface needs to be linked to a profile with the NGMFCMDS=YES key/value pair on its AUTH tag.

## DSICRTTD

The focal points need to be identified to your target systems. Uncomment and adapt the following lines for any of your target systems:

```
*  DEFFOCPT PRIMARY=CNM02LUC,TYPE=ALERT,BACKUP=CNM99LUC
*  alerts
*  RMTCMD/XCF
```

## DUIFPMEM

Uncomment and adapt the following 4 statements.

```
*USETCPIP = NO
*TCPANAME = &CNMTCPN
*SOCKETS  = 50
*PORT = 4020
```

Change USETCPIP to YES. Change the PORT number to an unused number in your system if necessary.

### DUIGINIT

Change the domain specification to your focal point domain.

If you use Kanji support check that GMFHS is enabled to send Japanese text to an NMC console for display. In DUIGINIT you have to set JAPANESE=ON.

### INGTOPOF

Define your sysplex to your NMC as described in "Step 26D: Customize the INGTOPOF File."

## Step 26C: Customize RODM

You need to configure RODM so that it will dynamically refresh the workstation when a number of fields other than DisplayResourceStatus is changed. To do this you need to ensure that certain RODM loader statements are processed whenever the GMFHS Data Model is reloaded.

Add the DD statement with member INGDYNRF in the NetView sample procedure EKGLOADP.

```
  ⋮
//*EKGIN1   DD DSN=&EKGIN1,DISP=SHR
//EKGIN1 DD DSN=&SQ1..V&NETVER..CNMSAMP(DUIFSTRC),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM1),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM2),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM3),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM4),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM5),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM6),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM7),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM8),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDM9),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDMA),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDMB),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDMC),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDMD),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDME),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDMF),DISP=SHR
//        DD DSN=&SQ1..V&NETVER..CNMSAMP(FLBTRDMZ),DISP=SHR
//*  Dynamic update of resources
//     DD DSN=&SQ2..V&SAMVER..SINGSAMP(INGDYNRF),DISP=SHR
//*
//*
  ⋮
```

*Figure 21. Sample of RODM Load Procedure EKGLOADP*

## Step 26D: Customize the INGTOPOF File

The generic name for the topology control file is "INGTOPOF". Local versions of the INGTOPOF file may also be created.

The naming (format) of the local versions will be "TPF" concatenated with the domain name of the focal point. For example, if the focal point has a domain name of "IPSNM", the local INGTOPOF name will be "TPFIPSNM".

Multiple INGTOPOF files (generic and local) may exist with a single DSIPARM. This will provide the flexibility to tailor each INGTOPOF to suit the requirements of each focal point.

When the topology manager attempts to read the topology control file, in the first instance it will look for the local INGTOPOF member name in DSIPARM. Processing is as follows:

1. If the local INGTOPOF member exists in DSIPARM, the content of that member will be used by the topology manager.
2. If the local INGTOPOF member does not exist in DSIPARM, the topology manager will attempt to read the INGTOPOF member in DSIPARM.
3. If the INGTOPOF member exists in DSIPARM, the content of that member will be used by the topology manager.
4. If the INGTOPOF member does not exist in DSIPARM, the topology manager will terminate with RC = 9.

The following overview of the operation mode of the SA z/OS topology manager supplies some background for discussing the INGTOPOF file. Some familiarity with the class structure of RODM and with the BLDVIEWS tool is assumed.

During initialization, the SA z/OS topology manager gathers information about generated SA z/OS resources from the sysplex and stores the resources in RODM, prefixing their names with the current sysplex name. Usually not only the resources, but also the dependencies and major/minor relationships between resources will be represented in RODM (this depends on the OPTION statement in the INGTOPOF file, see Appendix B, “Syntax for INGTOPOF File,” on page 183).

The INGTOPOF file supplies the SA z/OS topology manager with the following information:

- which sysplexes there are and which of their member systems contain a SA z/OS topology agent.
- the names of the data sets (members) that contain the definitions of the views.
- when views must be rebuilt during runtime, it is desirable that only those views be rebuilt to which new members have been added.

You will need to prepare the INGTOPOF input file. This contains information about the target domains and how they are grouped into sysplexes along with some additional information that affects the resources that are dynamically created.

The INGTOPOF file contains configuration information for the SA z/OS topology manager. It must reside in DSIPARM. The records of the file consist of a keyword with one or more parameters. Comment lines must have an asterisk (*) in the first column. A ′+′ at the end of a line indicates that the record is continued in the next line.

The information is passed from the INGTOPOF file to the SA z/OS topology manager with the help of the following keywords:

- SYSPLEX
- PROCOPS
- BLDVIEWS
- [LOCATION]
- [ANCHOR]
- [OPTION]
- [TEMPLATE]
- [MAPCOLOR]

The syntax of the statements in the INGTOPOF file is described in Appendix B, "Syntax for INGTOPOF File," on page 183.

A sample of INGTOPOF is provided in the SINGNPRM library.

To start the MultiSystem Manager and load the INGTOPOF file, use the MultiSystem Manager start command FLCAINIT.

### Step 26E: Prepare BLDVIEWS Cards

You need to provide files with BLDVIEWS cards. These are required for the SA z/OS resources to appear on the NMC workstation. These files will become part of the BLDVIEWS statement in the INGTOPOF file. The BLDVIEWS statement in the INGTOPOF file is used by the SA z/OS topology manager to pass information to the BLDVIEWS tool which it invokes to produce the views of the objects. The BLDVIEWS tool writes information about views into RODM. The SA z/OS topology manager is automatically invoked whenever you start SA z/OS or you can invoke it with the INGTOPO command whenever you changed information in the INGTOPOF file or in the files with the BLDVIEWS cards.

To run the BLDVIEWS tool, use one of the following methods:

- via the SA z/OS topology manager which invokes the tool
- via an external invocation of this tool (as a NetView command in a NetView session)

For information about the BLDVIEWS cards syntax refer to the appropriate NetView documentation.

The following three SA z/OS BLDVIEW samples are provided in the SINGNPRM library matching the INGTOPOF sample file:

- INGBVIEW (sample view for SysOps objects)
- INGPVIEW (sample view for ProcOps objects)
- INGCVIEW (sample view for common objects)

**Note:** To start MultiSystem Manager and load the INGTOPOF file, use the MultiSystem Manager start command: `FLCAINIT`

## Step 27: Copy and Update Sample Exits

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| * | * | * |

Several sample exits are provided in the SINGSAMP library (for example, AOFEXC01). You can use these samples to create your own exits. If used, they must be copied into a data set (either the enterprise-specific or domain-specific) in the DSICLD concatenation. These exits are called at fixed points during SA z/OS processing. Therefore, you should look into each of the sample exits to determine whether you need to use and update it.

Updating and copying the sample exits allows you to add your specific processing. For more information on user exits, provided samples and advanced automation options, refer to *IBM Tivoli System Automation for z/OS Customizing and Programming*.

# Step 28: Install CICS Automation in CICS

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| * | | |

This section describes the basic CICS Automation definitions that take place on CICS. Refer to the CICS documentation while performing these steps, especially the *CICS Resource Definition Guide*. These steps are performed on each CICS region.

**Note:** The TS queues, EVEVCQUE and COLEEVEQ, that are used by SA z/OS CICS must not be defined as remote in your temporary storage table (TST).

## Step 28A: SIT or Startup Overrides

On each CICS, ensure that the system initialization table (SIT) or startup overrides include the following:

```
PLTPI=xx,            where xx is the suffix to the startup PLT
PLTSD=yy,            where yy is the suffix to the shutdown PLT
MSGLVL=1,
BMS=(STANDARD|FULL)
```

Because CICS Automation maintains a long-running task in each CICS, review the AMXT, CMXT, and MXT values.

You may optionally add CN as your last startup override, whether from SYSIN or through the JCL. However, this is not necessary if you have added the &APPLPARMS variable to the PARM of the CICS start command in the STARTUP item of the APPLICATION policy object. The following is an example:

```
MVS S cics,...,PARM='SYSIN,START=xxxx&APPLPARMS'
```

This is also how the start commands are predefined in the sample databases.

## Step 28B: Program List Table Definitions

Add the TYPE=ENTRY definitions shown in the following example to the post initialization program list table (PLT) for each CICS after the entry for DFHDELIM (as in phase 2).

```
DFHPLT TYPE=INITIAL,SUFFIX=xx
DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
DFHPLT TYPE=ENTRY,PROGRAM=EVEPYINI
DFHPLT TYPE=ENTRY,PROGRAM=EVESTISP
DFHPLT TYPE=FINAL
```

Add the TYPE=ENTRY definitions shown in the following example to the shut down program list table (PLT) for each CICS.

```
DFHPLT TYPE=INITIAL,SUFFIX=yy
DFHPLT TYPE=ENTRY,PROGRAM=EVESPLTT
DFHPLT TYPE=ENTRY,PROGRAM=EVESYLMQ
DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
DFHPLT TYPE=FINAL
```

Assemble the PLT tables.

## Step 28C: Define Consoles

CICS Automation uses EMCS consoles to issue Modify CICS commands when managing CICS. Console definitions are required for correct CICS Automation operation.

Define consoles for autotasks to enable CICS Automation functions. This step can be skipped if you enable CICS Auto-Installed Consoles. This can be achieved by specifying ″AICONS=YES″ in the CICS system initialization parameters.

In an EMCS environment the autotask console names are determined, in order of precedence as follows:

1. If you are using AOCGETCN (that is, using the profiles shipped with the product) the name is determined by AOFCNMASK. For more information, see *IBM Tivoli System Automation for z/OS Customizing and Programming* or *IBM Tivoli System Automation for z/OS Defining Automation Policy*.
2. The CONSNAME parameter on the PROFILE statement in the task profile determines the EMCS console name. For more information, see *Tivoli NetView for z/OS Administration Reference* and *Tivoli NetView for z/OS Security Reference*.
3. By default the autotask name is used for the EMCS console name.

A console has to be defined for each SA z/OS work operator. These are typically named AUTWRK*xx*. In addition, a console has to be defined for each NetView operator that may want to inquire or control a CICS region. This can be simplified by specification of the CICS Console Auto-Install function.

RACF security is provided by z/OS for EMCS and MCS consoles. This function enables a user on NetView with a RACF user ID (ACEE) to open an EMCS console and have the user ID associated with the EMCS console. All commands that are issued to the EMCS console will have the user ID of the NetView user. Furthermore, CICS supports EMCS and MCS consoles with RACF user IDs by inheriting the user ID that is associated with a command from the EMCS or MCS console.

The net result is that for CICS auto-installed consoles, the user ID that is assigned to the console is the user ID that issued the command. In the case of SA z/OS this would be the NetView user's user ID (only if NetView is using RACF to verify user IDs). This means that all tasks in NetView that require consoles will also require RACF user IDs and the appropriate permissions in CICS. This includes all human operators and all auto operators.

For those users who want to have a predefined user ID instead of the all the possible user IDs from NetView, the Console Model Terminal definition should specify a user ID in its definition.

## Step 28D: Transaction and Program Definitions

This step describes how to define the standard CICS Automation transactions and programs to CICS. To this purpose, the DFHCSDUP program is used.

The members required to run these jobs are provided with CICS Automation. However, some modifications are required, as described below:

**Hint**

You might want to back up your CSDs before doing this step.

For each CSD, run the EVESJ015 sample job. This job defines transactions and programs for CICS automation in four groups:

- EVEGRP1
- EVEGRP2
- EVEGRP3
- EVEGRP4

Before you run it, modify the job as directed in the JCL comments. The definitions must be updated for every CICS CSD file that has CICS subsystems that are to be controlled by SA z/OS. If there are existing CSD definitions for older releases of SA z/OS, run the sample job for this release to upgrade the definitions to the latest release. The definitions are downwards compatible to previous releases.

### Step 28E: DFHRPL and the CICS Automation Library

Update the DFHRPL concatenation to add the ING.SINGMOD1 library for every CICS subsystem that is to be managed by SA z/OS.

**Note:** Do ***not*** add these libraries to the DFHRPL for CICSPlex CMAS subsystems.

### Step 28F: Add Libraries for NetView

Uncomment any libraries that you require in the INGENVSA member of the SINGSAMP data set. Refer to the sample for more details.

## Step 29: Install IMS automation in IMS

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| * | | |

### Step 29A: Specify Required Control Region Parameters

Modify all IMS Control region and IMS DB control region JCL to specify the following parameter:

**CMDMCS=Y**
This is required for correct operation of IMS product automation.

**PREMSG=N**
This is required for correct operation of IMS Product Automation.

### Step 29B: Install DFSAOE00 Exit

There are three ways to install the exit.

- Use the default z/OS exit router as supplied by SA z/OS.
  - This involves concatenating the ING.SINGMOD1 library before the IMS.SDFSRESL library in the STEPLIB concatenation.

  - Add PROG*xx* members to SYS1.PARMLIB to define the exit. Sample member EVISI005 contains the base required definitions. See *IBM Tivoli System Automation for z/OS IMS Automation Programmer's Reference and Operator's Guide* for further customization details.
- Use the SA z/OS-supplied exit on its own.
  - This involves concatenating the ING.SINGMOD1 library after the IMS.SDFSRESL library in the STEPLIB concatenation, unless ING.SINGMOD1 is in the linklist concatenation chain.
  - Relink the EVIPVEX1 module and give it an ALIAS of DFSAOE00 into a library concatenated before IMS.SDFSRESL in the STEPLIB concatenation. Sample EVISJ001 is an example of how to do this.
- Call the SA z/OS exit from your routine.
  - This involves concatenating the ING.SINGMOD1 library after the IMS.SDFSRESL library in the STEPLIB concatenation, unless ING.SINGMOD1 is in the linklist concatenation chain.
  - Call the EVIPVEX1 module from your exit program as detailed in *IBM Tivoli System Automation for z/OS IMS Automation Programmer's Reference and Operator's Guide*.

## Step 30: Install TWS Automation in TWS

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| * | | |

### Step 30A: Add Libraries to TWS

Add your SINGMOD1 library and the NetView CNMLINK library containing CNMNETV to the TWS steplib. Alternatively, you may add these libraries to LINKLST. You should have already APF-authorized these libraries.

### Step 30B: Add Libraries to NetView

Allocate the EQQMLOG library according to your TWS definitions. This data set contains any error messages that may occur when using the TWS APIs on this NetView.

EQQMLIB should point to the appropriate message library for the level of TWS that you are running

Uncomment any libraries that you require in the INGENVSA member of the SINGSAMP data set. Refer to the sample for more details.

### Step 30C: Update TWS Parameters and Exits

Install the exit module EEQUXSAZ. This exit is required for TWS Automation workstation processing.

A recycle of TWS is required to install the exit 7 module EQQUX007 or the exit 11 module EQQUX011. If you are using an existing exit 7 or exit 11, you can combine these exits with modules that are supplied by TWS Automation.

TWS Automation supplies EQQUX007 to detect workstations that are used for NetView communication. The following modules are used as part of this process:

```
EQQUX007
UX007001
UX007004
EQQUX011
UX011011
```

EQQUX007 and EQQUX011 are the exit driver programs. They call other modules in turn, as though TWS is calling each module directly.

The EQQUX007 driver searches for UX007001 through UX007010, and the EQQUX011 driver searches for UX011001 through UX011010. UX007001, UX007004, and UX011001 are supplied with TWS Automation.

If you have an existing exit 7, rename your module from EQQUX007 to UX007005. If you have an existing exit 11, rename your module from EQQUX011 to UX011002.

The called routines are passed the same parameters as the call to EQQUX007 or EQQUX011.

If you want to add additional exit 7 or exit 11 modules, use the next available name, such as UX007005 or UX011002. This makes it easier to integrate exits that are supplied by various products. Also, because modules are loaded dynamically by the exit driver on each invocation, you may add, delete, or modify an exit module without recycling TWS.

You must specify the CALL07(YES) parameter in the TWS/ESA initialization parameters.

You must specify the CALL11(YES) parameter in the TWS/ESA initialization parameters if you want to monitor CP deletes. CP delete monitoring allows TWS Product Automation to clear outstanding SDF and NMC alerts when an application or operation is deleted from the current plan.

Other initialization parameters must be specified in the TWS initialization member (EQQPARM) so that TWS will issue some of its messages to the MVS console.

The DURATION, ERROROPER, LATEOPER, and OPCERROR messages are automated by TWS Automation. The RESCONT and QLIMEXCEED messages are useful for further customer automation.

You must specify the following in EQQPARM:

```
ALERTS WTO (DURATION
   ERROROPER
   LATEOPER
   RESCONT
   OPCERROR
   QLIMEXCEED)
```

In addition, you must edit the TWS-supplied message members for certain messages.

The following messages are automated and may require changes to the TWS-supplied message members in the SEQQMSG0 data set:

| **Message** | **Member** |
|---|---|
| EQQE026I | EQQE02 |
| EQQE036I | EQQE03 |
| EQQE037I | EQQE03 |
| EQQE107I | EQQE10 |
| EQQFCC1I | EQQFCC |
| EQQN013I | EQQN01 |
| EQQPH00I | EQQPH0 |
| EQQW011I | EQQW01 |
| EQQW065I | EQQW06 |
| EQQW079W | EQQW07 |
| EQQZ006I | EQQZ00 |
| EQQZ086I | EQQZ08 |
| EQQZ128I | EQQZ12 |
| EQQZ200I | EQQZ20 |
| EQQZ201I | EQQZ20 |

Modify these message members to include WTO=YES for the indicated message IDs. Full details for customizing TWS can be found in *Tivoli Workload Scheduler for z/OS Customization and Tuning*.

**Note:** If you use NMC and SDF to monitor the status of TWS operations, you should enable UX007004 and update INGMSGU1 to remove the Message Automation traps for EQQE026I and EQQE036I. This is to prevent you from receiving multiple NMC and SDF alerts for the same TWS event as a result of the following:

- NMC and SDF alerts that are generated from EQQE036I do not contain an operation number. Therefore, if an application contains operations that have identical job names (with the same IATIME and same workstation ID), it is possible that duplicate or ambiguous alerts are generated.
- Alerts that are generated from EQQE026I and EQQE036I are not removed from NMC and SDF if UX007004 is not active. This is because TWS does not issue a message when these operations exit error status.

# Step 31: Install USS Automation

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| * | | |

## Step 31A: Define UNIX Segments (OMVS)

Depending on the NetView operator security definition, one or more UNIX segments must be defined. These OMVS segments can have a root UID (0) or a non-root UID. To run a non-root UID requires more setup.

**When using OPERSEC=MINIMAL, NETVPW, or SAFPW**, one OMVS segment must be defined. This is the segment for the user ID running NetView.

**When using OPERSEC=SAFCHECK, or SAFDEF** (user level security), the following operator IDs need a UNIX segment:

- AUTWRK01-NN
- RPCOPER
- MONOPER
- AUTRPC
- AUTO1
- AUTSYS (backup task for AUTRPC and AUTO1)
- AUTBASE (backup task for AUTRPC and AUTO1)
- All tasks that receive actions from the AT for UNIX resources. Usually these are the work operators.

### Using the OMVS Segment with Root UID

This is the easiest way to set up the z/OS UNIX segment. Giving it a UID of 0 (root user) enables this user to operate without restrictions. This segment must also be permitted to the RACF facility class BPX.DAEMON (if defined).

**Note:** Any user that can change NetView common global variables may be able to issue UNIX System Services commands under a root user ID.

### Using the OMVS Segment with Non-Root UID

If you want to reduce the number of UID 0 users, it is possible to define a setup without UID 0 with some restrictions.

If you are using a setup with non-root UID, the OMVS segment must be defined in the following way:

**Monitoring:**

- For process monitoring:

  Define read access to `SUPERUSER.PROCESS.GETPSENT`

  This allows a user ID to see all processes. If the user ID performing the monitoring is not allowed to check all processes, the automation may assume that the start was not successful and restarts the application. This will result in many instances.
- For file or filesystem monitoring:

  Define read access to `SUPERUSER.FILESYS`

This allows a user ID to get access to all files in the UNIX file system. If the user ID performing the monitoring is not allowed to check all files, the automation may assume that the resource is unavailable.

- Give access to any resource that user-written monitoring routines may use
- For user-defined monitoring, see "Command Execution (INGUSS)" below. (User defined monitoring is performed with the command INGUSS.)

**Command Execution (INGUSS):**

- Give the OMVS segment the ability to switch to any user ID associated with z/OS UNIX resources (access to `BPX.SRV.userid` or `BPX.SUPERUSER` to start root programs).
- Depending on your security environment the OMVS segment may need access to `BPX.DAEMON`.
- The OMVS segment must be authorized to perform all the commands that are specified in the customization dialogs. For an overview of authorizations for non-root users, refer to the chapter that explains UNIXPRIV class profiles in *z/OS UNIX System Services Planning*.

**Restrictions for Non-Root UID Setup:** There is an MVS identity and an z/OS UNIX identity. Without a UID 0 you cannot switch the MVS identity. If a user needs access to certain MVS data sets, you may not start the application with INGUSS. You may have trouble when automating z/OS UNIX resources that require a UID of 0 (for example, the inetd). The OMVS segments without UID 0 are normally not able to switch to a root user in order to perform actions. SA z/OS standard monitoring will work. For example, if you allow the OMVS segment to switch to UID 0 (by defining read access to BPX.SUPERUSER), you could also assign it a UID of 0.

## Creating an OMVS Segment by Submitting a Job

Creating OMVS segments can be done by submitting a job, as shown in Figure 22 on page 153.

The NOPASSWORD option prevents unauthorized logins.

This OMVS segment must be authorized to set the jobname (read access to `BPX.JOBNAME`). Otherwise, the started address spaces have the same jobname as NetView. When the jobname can be set, the newly created address space has the jobname INGCUNIX.

If the started UNIX processes are to have a user-defined MVS jobname (specified with the `JOBNAME` parameter of the INGUSS command), the target user IDs that are issuing the commands must have at least read access to RACF facility class `BPX.JOBNAME`. Otherwise, a jobname will be assigned by the operating system. The target user ID is the user that this resource is assigned to in the customization dialog panel, z/OS UNIX Control Specification.

```
//*
//ADDUSER  EXEC PGM=IKJEFT01
//*
//SYSTSPRT DD SYSOUT=*
//SYSLBC   DD  DSN=SYS1.BRODCAST,DISP=SHR
//SYSTSIN  DD *
  ADDUSER STCUSER +
          NOPASSWORD+
          UACC(NONE) DFLTGRP(AUTGRP) +
          OMVS(UID(0000000) HOME('/')  PROGRAM('/bin/sh')) +
//*
//COUSERS  EXEC PGM=IKJEFT01
//*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  CO      STCUSER GROUP(USERS) AUTH(USE)
//*
```

*Figure 22. Job Example of Creating an OMVS Segment*

## Step 31B: Preparing for USS Automation

Use the common global variable, AOFUSSWAIT, that you can set in your startup exit, to change the way SA z/OS behaves. This variable should be set only once for an SA z/OS system.

AOFUSSWAIT is the time that SA z/OS waits for the completion of a user-specified z/OS UNIX monitoring routine (defined in the z/OS UNIX Control Specification panel) until it gets a timeout. When the timeout occurs, SA z/OS does no longer wait for a response from the monitoring routine and sends a SIGKILL to the monitoring routine.

# Step 32: Customizing GDPS

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| * | | |

This section describes the necessary customization and definitions when running GDPS on top of SA z/OS.

| You can also import the sample add-on policy, *GDPS, which is delivered with SA z/OS, into your policy database and customize its definitions there to fit your environment.

## Step 32A: Preparing NetView

1. Concatenate the SGDPPARM product data set to the DSIPARM DD-statement in the NetView startup procedure. See the INGENVSA sample that is provided by SA z/OS in the SINGSAMP library for more details.
2. If you need to modify the INGXINIT member, which is the initialization member of the SA z/OS communication task for the production system or its equivalent, INGXKSYS, for the GDPS controlling system, copy them to your user data sets and make your modifications there.

   INGXKSYS uses the z/OS system symbol &SYSCLONE. as the XCF group ID. This allows the same member to be used for all controlling systems. The resulting XCF group will always be created in a unique way: INGXSG*xx*, where *xx* is the value of &SYSCLONE. This corresponds to HSAPRMKS as described in "Step 32B: Preparing the Automation Manager."

3. If necessary, copy the INGSTGEN member from the sample library (SINGSAMP) to the CNMSTGEN member of the DSIPARM data set of each NetView instance in your sysplex and adapt the TOWER statements according to your installation.

   Additionally specify the GDPS product of your installation and whether this is the GDPS controlling system (KSYS) or production system (PROD) by removing the asterisk in front of the appropriate line:

   `*TOWER.SA.GDPS=PPRC KSYS`
   : If GDPS/PPRC is installed and this is controlling system

   `*TOWER.SA.GDPS=PPRC PROD`
   : If GDPS/PPRC is installed and this is production system

   `*TOWER.SA.GDPS=HM KSYS`
   : If GDPS/PPRC HM is installed and this is controlling system

   `*TOWER.SA.GDPS=HM PROD`
   : If GDPS/PPRC HM is installed and this is production system

   `*TOWER.SA.GDPS=XRC`
   : If GDPS/XRC is installed for all systems, SA z/OS will initialize all systems with INGXINIT

   `*TOWER.SA.GDPS=GM`
   : If GDPS/GM is installed for all systems, SA z/OS will initialize all systems with INGXKSYS

   **Note:** If the TOWER.SA statement includes GDPS, the VPCEINIT installation exit that is required by each supported GDPS product is automatically called during initialization of SA z/OS. You no longer need to specify it in each system's SYSTEM INFO policy in the customization dialog.

## Step 32B: Preparing the Automation Manager

The GDPS controlling system must run in a separate XCF group (subplex) and therefore has its own automation manager. The automation manager parmlib member for the controlling system (K-system) is HSAPRMKS, using the z/OS system symbol &SYSCLONE as the XCF group ID. This allows the same parmlib member to be used for all controlling systems. The resulting XCF group will always be created in a unique way: INGXSG*xx*, where *xx* is the value of &SYSCLONE.

Copy and edit the automation manager startup procedure INGEAMSA. The same startup procedure can be used for the automation manager that controls the production systems and the automation manager that controls the K-system, assuming that the PARMLIB member suffix is specified on invocation of the procedure.

## Step 32C: Defining the Automation Table Used by GDPS

SA z/OS provides a NetView automation table that contains all the messages that are required by GDPS. The relevant AT is loaded, depending on the specified GDPS Tower statement, as follows:

| Tower Statement | AT loaded |
|---|---|
| TOWER.SA.GDPS=PPRC | GEOMSGGP |

| Tower Statement | AT loaded |
|---|---|
| TOWER.SA.GDPS=HM | GEOMSGHM |
| TOWER.SA.GDPS=XRC | GEOMSGXR |
| TOWER.SA.GDPS=GM | GEOMSGGM |

Add the INGMSGGP automation table in the system policy so that the table is automatically loaded by SA z/OS at initialization time.

## Step 33: Customizing I/O Operations

| SysOps | ProcOps | I/O Ops |
|---|---|---|
|  |  | ✓ |

With z/OS 1.8, MVS introduced the new parameter "VSM ALLOWUSERKEYCSA(NO|YES)" in the DIAG*xx* parmlib member. The default value for z/OS 1.8 is YES, whereas from z/OS 1.9 onwards the default value is NO. However, I/O Operations requires a value of YES for the following reason: All I/O Operations routines use preallocated stack storage for performance reasons. Because some exit routines are invoked in user key but outside the I/O Operations address space, the stack storage that is being used by these routines must be allocated in user key.

**DIAGxx**

I/O Operations requires the following setting:

```
VSM ALLOWUSERKEYCSA(YES)
```

This is the default when the parameter is not specified.

This section describes optional customization when running I/O operations.

```
//IOOPS    EXEC PGM=IHVOINI,    **IHV INITIALIZATION MODULE NAME**
//             TIME=1440,       **RUN FOREVER**
//             REGION=4M,       **REGION SIZE**
//             DPRTY=(15,15),   **PRIORITY OF TASK**
//             PARM=''
//*
//STEPLIB  DD  DISP=SHR,DSN=#hlqinst#.SINGMOD1
//HCDTRACE DD  DISP=SHR,DSN=#hlq#.&SYSNAME..HCDTRACE
//*HCDPROF  DD  DISP=SHR,DSN=#hlq#.HCDPROF
```

*Figure 23. Startup JCL of I/O operations*

You may add the PARM parameter:

`TIMEOUT=0–999999`
: This sets the timeout value of the very first I/O operations application to the specified value. All other I/O operations applications ignore the parameter because they inherit the timeout value from the first running I/O operations application that the new application communicates with.

`COMM={TCP|VTAM}`
: This restricts communication to TCP/IP or VTAM only.

**Note:** Running a mix of I/O operations applications, some started with COMM=TCP and some with COMM=VTAM, leads to unpredictable results.

COMM=TCP should only be specified when ***all*** I/O operations applications run SA z/OS V3.2 or higher.

TPNAME=*tpname*
This specifies the TCP/IP procedure name that is used for communication.

In a multi-stack environment (CINET) TCP/IP allows up to 8 different address spaces running in parallel on a single MVS image. This parameter restricts the TCP/IP communication of I/O operations to the specified procedure. Otherwise the first active procedure is used.

**Note:** Do not specify the parameter in a single-stack environment (INET).

Two or more parameters must be separated with a comma.

## Step 34: Installing Tivoli Enterprise Portal Support

| SysOps | ProcOps | I/O Ops |
|---|---|---|
| * | | |

If you plan to use the SA z/OS monitoring agent you must perform the SMP/E installation of the support for the Tivoli Enterprise Portal (TEP). For further details, refer to *IBM Tivoli System Automation for z/OS Monitoring Agent Configuration and User's Guide* and the *IBM Tivoli Monitoring Services: Program Directory*.

You can import the sample add-on policy, *ITM, which is delivered with SA z/OS, into your policy database and customize its definitions there to fit your environment.

# Chapter 9. Installing SA z/OS on Workstations

This chapter contains information on how to install those parts of SA z/OS that are required on workstations:

- “Installing the NMC Workstation”
- “Installing and Customizing the TEC Event Server Workstation” on page 164

The workstation components can be installed on any workstation that meets the requirements listed in Chapter 1, “SA z/OS Prerequisites and Supported Equipment,” on page 3. One or more workstations can be installed for users to monitor and control the systems being managed with SA z/OS.

The code for the SA z/OS NMC exploitation is supplied with the host code that is installed using SMP/E. Installing the SA z/OS NMC exploitation will enable you to issue the most important SA z/OS processor operations and system operations commands from all NMC workstations.

**Note:** The NMC installation described in “Installing the NMC Workstation” is performed on the NMC Server and the NMC clients. After this installation, you need to restart the individual NMC clients.

## Installing the NMC Workstation

If you already have an NMC environment installed, you can continue with the actions described in the remainder of this section. Having completed these, you can use the SA z/OS NMC exploitation as described in *IBM Tivoli System Automation for z/OS User's Guide*. This will enable you to issue a selection of SA z/OS processor operations and system operations commands from all NMC workstations.

The following packed files for the SA z/OS NMC exploitation are available after your SMP/E installation:

- ING.SINGPWS1(INGNMCZP):

  INGNMCZP is the packed file for Windows. Download it with the extension ZIP and unpack with an appropriate tool (WINZIP or PKZIP).
- ING.SINGPWS1(INGNMCTZ):

  INGNMCTZ is the SA z/OS workstation code for AIX, UNIX and z/Linux workstations.

  Step 1. Download the member (INGNMCTZ) from the data set on the host system to your workstation in *binary* mode using, for example, FTP.

  Step 2. Rename INGNMCTZ to INGNMCTZ.tar.gz on the workstation.

  Step 3. Uncompress with the command:

  ```
  gzip --decompress --verbose INGNMCTZ.tar.gz
  ```

Step 4. Unpack with the command:

```
tar --extract --verbose --file=INGNMCTZ.tar
```

This creates the subdirectory INGNMCEX on the workstation.

- ING.SINGPWS1(INGNMCZJ): Japanese version of the packed file for Windows workstations

  If you use the Japanese version of SA z/OS download this file with extension ZIP and unpack with an appropriate tool (WINZIP or PKZIP).
- ING.SINGPWS1(INGNMCTJ): Japanese version of the packed file for UNIX workstations

  If you use the Japanese version of SA z/OS download this file with extension TAR.Z and unpack and uncompress with an appropriate tool (*uncompress* and *tar*).

The content of each packed file is divided into a support for system operations commands and a support for processor operations commands. Both packages include two NMC response files. One response file contains the system operations commands, the other one contains the processor operations commands. The response files include the definitions and profiles for

**ING_SO_OPER**
SystemOperation Operator

**ING_PO_OPER**
ProcessorOperation Operator

**ING_SA_OPER**
SystemAutomation Operator (definition for both the system operations and processor operations commands)

Furthermore there are two subdirectories for the related data definition files and two subdirectories with the online help in HTML format.

With this separation of system operations and processor operations commands you may install either the system operations commands or the processor operations commands or both depending on your needs. The installation has to be done manually, as there is no common installation tool for the several supported platforms. This requires that you are familiar with the common commands of your workstation operating system.

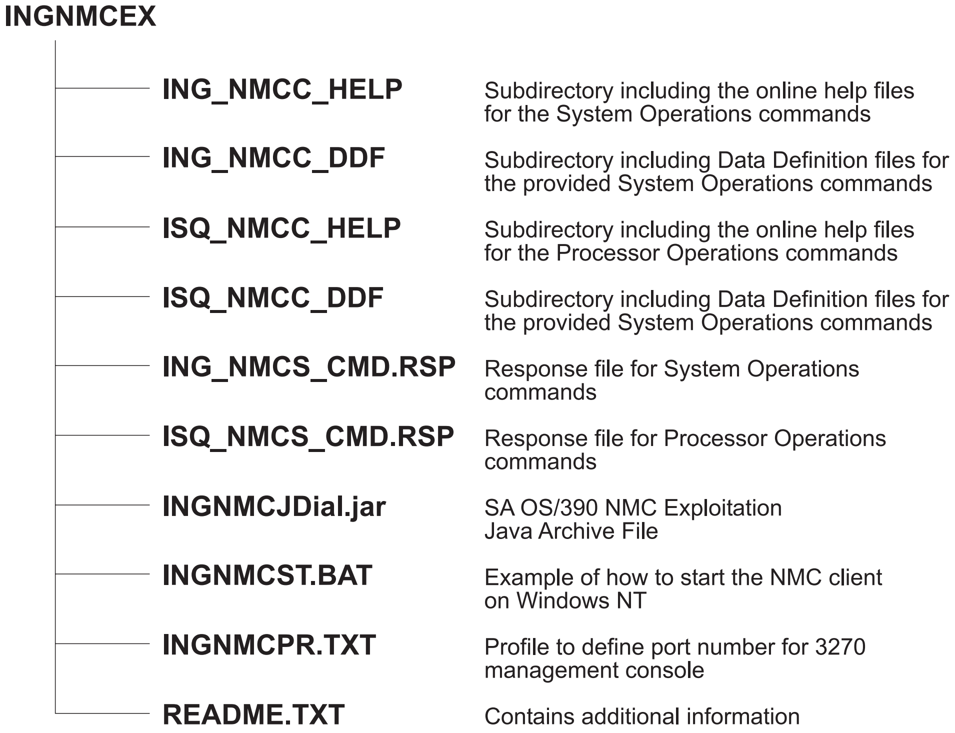

INGNMCEX
ING_NMCC_HELP
Subdirectory including the online help files for the System Operations commands
ING_NMCC_DDF
Subdirectory including Data Definition files for the provided System Operations commands
ISQ_NMCC_HELP
Subdirectory including the online help files for the Processor Operations commands
ISQ_NMCC_DDF
Subdirectory including Data Definition files for the provided System Operations commands
ING_NMCS_CMD.RSP
Response file for System Operations commands
ISQ_NMCS_CMD.RSP
Response file for Processor Operations commands
INGNMCJDial.jar
SA OS/390 NMC Exploitation Java Archive File
INGNMCST.BAT
Example of how to start the NMC client on Windows NT
INGNMCPR.TXT
Profile to define port number for 3270 management console
README.TXT
Contains additional information

*Figure 24. Directory Structure of Unpacked Files*

## Installation Steps on the NMC Server

Perform the following steps to install SA z/OS NMC exploitation on the NMC Server (it should be noted, the term UNIX in the following steps refers to all forms of UNIX derivatives, including AIX, z/Linux, etc.):

1. Download the appropriate packed file in binary format to the NMC Server.
2. Unpack the file into a temporary directory of the NMC Server, using an appropriate tool for the NMC Server operating system. You will obtain the directory structure for the unpacked files as shown in Figure 24.
3. Copy the required help files as follows:

| Environment | From Directory | To Your Directory |
|---|---|---|
| WIN | `tmp\INGNMCEX\ING_NMCC_HELP` and/or `tmp\INGNMCEX\ISQ_NMCC_HELP` | `[BINDIR]\TDS\server\db\current\help` |
| UNIX | `tmp/INGNMCEX/ING_NMCC_HELP` and/or `tmp/INGNMCEX/ISQ_NMCC_HELP` | `$BINDIR/TDS/server/db/current/help` |

   where *tmp* stands for the directory where you downloaded the files.

**Note:** ***BINDIR*** is an environment variable set by your NMC installation and indicates that this is a subdirectory of your installed NMC product. For example:

```
usr\local\Tivoli\bin\w32-ix86\
```

4. Copy the required data definition files as follows:

| Environment | From Directory | To Your Directory |
|---|---|---|
| WIN | *tmp*\INGNMCEX\ING_NMCC_DDF and/or *tmp*\INGNMCEX\ISQ_NMCC_DDF | [BINDIR]\TDS\server\config\ddf\c |
| UNIX | *tmp*/INGNMCEX/ING_NMCC_DDF and/or *tmp*/INGNMCEX/ISQ_NMCC_DDF | $BINDIR/TDS/server/config/ddf/c |

5. Copy the required response files from INGNMCEX as follows:

| Environment | To Your Directory |
|---|---|
| WIN | [BINDIR]\TDS\server\sample |
| UNIX | $BINDIR/TDS/server/sample |

6. Copy the Java™ archive file INGNMCJDial.jar from INGNMCEX as follows:

| Environment | From Directory | To Your Directory |
|---|---|---|
| WIN | *tmp*\INGNMCEX | [BINDIR]\TDS\server\db\current\lib |
| UNIX | *tmp*/INGNMCEX | $BINDIR/TDS/server/db/current/lib |

7. Verify the following:
   a. To operate the NMC Server you must be logged on to NetView via a 3270 host session.
   b. Your NetView user ID must have NGMF administrator rights.
   c. The NMC Server must be started and active.
   d. The connection from the NMC Server to NetView must be established.
8. Start the *Command Profile Editor batch utility* (CPEBATCH) with:
   a. for WIN environment
      - [BINDIR]\TDS\server\sample\ING_NMCS_CMD.RSP and/or
      - [BINDIR]\TDS\server\sample\ISQ_NMCS_CMD.RSP

      and the -i and -g parameters
   b. for UNIX environment
      - $BINDIR/TDS/server/sample/ING_NMCS_CMD.RSP and/or
      - $BINDIR/TDS/server/sample/ISQ_NMCS_CMD.RSP

      and the -i and -g parameters

   With this step, you load the delivered commands into the NetView internal database. For information on how to use this batch utility, refer to *NetView Management Console User's Guide*. For a detailed description of how to maintain and manipulate response files for the NMC topology server, go to the SA z/OS Web page at

   http://www.ibm.com/servers/eserver/zseries/software/sa/adds/hint03.html
9. Use the Command Profile Editor batch utility (CPEBATCH) to apply the new profiles installed with step 8 to the individual operators defined in your installation. Just these operators that are linked to one of the SA profiles can execute SA commands. All other operators cannot display or execute SA commands.

For more details on CPEBATCH refer to the appendix 'Topology Server Commands' in the manual *Tivoli NetView for z/OS: NetView Management Console User's Guide*.

**Notes:**

a. The NetView CPE online utility was retired with NetView 5.1. Installations that are running NetView 1.4 can still use the CPE online utility to modify the definitions. The CPE online utility was never available for UNIX installations.

b. The recommended way to maintain definitions like operators, profiles, etc. is to use the tool delivered in the INGRSPTOOL.ZIP file. The tool comes with a detailed description. It can be downloaded from the SA z/OS Web page at

   `http://www.ibm.com/servers/eserver/zseries/software/sa/adds/hint03.html`

## Installation Steps on the NMC Client

You must have the ***NetView 3270 Management Console*** installed if you want to use full screen commands. Refer to the *NetView Management Console User's Guide* for information on how to do this.

**Note:** You cannot use full screen commands when the NMC focal point is a satellite installation. Use line mode commands instead. More details can be found in the ingnmcex/readme.txt mentioned above.

1. Set the environment variable TCONSOLE_CLASSPATH:
   a. for WIN environments pointing to:

      `[NMC_Client_Installation_path]\TDS\client\lib\INGNMCJDial.jar`

   b. for UNIX environments pointing to:

      `[NMC_Client_Installation_path]/TDS/client/lib/INGNMCJDial.jar`

   Refer to Figure 25 on page 163 for a sample batch file.
2. On the individual NMC Clients: Restart your NetView Management Console to incorporate your changes.
3. Customize the NetView 3270 Management Console. Execute these steps only if you use full screen commands:
   a. On the NMC, select an SA z/OS resource from an existing view. For this resource, select an SA z/OS command that needs to be transferred to the NetView 3270 Management Console, for example, the INGVOTE_FS command. Click on INGVOTE_FS to display the NetView 3270 Management Console, that does not show any output yet.
   b. Select ***Session Services*** from the NMC menu bar, and choose ***Add/Delete/Modify Session*** from the menu items. This opens the ***Add/Delete/Modify Session*** window.
   c. In the ***Full Screen Session Name*** field of this window type: `SA`
   d. In the ***Start command String*** field type, for example: `window date`

      (You can enter any valid NetView command.)
   e. Select the radio button ***Immediate***
   f. From the ***Session Options*** select: ***Start Automatically***
   g. Press the ***Add*** push button, then the ***Save*** push button to save your changes
   h. Press the ***Done*** push button to exit this window
   i. In the NMC, select the added *SA* pull-down choice from the ***Session Services*** menu bar item

j. To verify the customization, issue the INGVOTE_FS command to display the desired output

## Sample to Start the NMC (for Windows NT Environment)

```
@rem **************************************************************************
@rem IBM System Automation for z/OS NetView Management Console Exploitation
@rem Sample Program - 5645-006
@rem               (C) Copyright IBM Corp. 2004
@rem                    All rights reserved.
@rem
@rem SAMPLE PROGRAM - NO WARRANTY EXPRESSED OR IMPLIED
@rem
@rem You are hereby licensed to use, reproduce, and distribute these sample
@rem programs as your needs require.  IBM does not warrant the suitability or
@rem integrity of these sample programs and accepts no responsibility for their
@rem use for your applications.  If you choose to copy and redistribute
@rem significant portions of these sample programs, you should preface such
@rem copies with this copyright notice.
@rem **************************************************************************
@rem
@rem PRODUCT          (System Automation for z/OS)
@rem COMPONENT        (NMC Exploitation)
@rem FIRST_RELEASE    (V2R1)
@rem LAST_CHANGE      (11Jan2002)
@rem
@rem MODULE_NAME      (ingnmcst.bat)
@rem DESCRIPTIVE_NAME (Start the NMC Topology Console)
@rem **************************************************************************
@rem
@rem Function:  This sample shows how the NMC Topology Console can be
@rem            started. This sample was written for the Windows NT environment
@rem            and NMC 1.3.0.1.
@rem
@rem Usage:
@rem
@rem - The following is a sample which will NOT properly work until customer
@rem   installation specific data is provided.
@rem
@rem - Adapt the drive and path statements to reflect your installation
@rem   environment.
@rem   This example assumes that the NMC Topology Console was installed on
@rem   drive E:.
@rem
@rem - A good location to put this file is the directory:
@rem   E:\usr\local\Tivoli\bin\generic_unix\TDS\client\bin
@rem   If it is necessary it can be stored anywhere else.
@rem
@rem - Call this file from a icon on your desktop or from Windows
@rem   Start-Programs-Netview-... pull-down or from the command line.
@rem **************************************************************************

@setlocal

@rem Changes the user's current working directory to the 'bin' directory in
@rem the "base" console installation path.
E:
cd E:\usr\local\Tivoli\bin\generic_unix\TDS\client\bin

@set TIVOLI=e:\usr\local\Tivoli\bin\generic_unix\Tds
@set INGJAR=\client\lib\INGNMCJDial.jar
@set FLBJAR=\ibmflb\jars\tivflb13.jar

set TCONSOLE_CLASSPATH=%TIVOLI%%FLBJAR%;%TIVOLI%%INGJAR%
tconsoleNT.bat     .. -key nmc
@endlocal
```

*Figure 25. Sample to Start the NMC (for WIN Environment)*

## Installing and Customizing the TEC Event Server Workstation

The TEC event server can run on either a UNIX or a Windows NT® workstation. The following example describes the installation on UNIX. For the Windows NT installation, use Windows NT command syntax.

1. Download the package file INGPTEC containing the workstation code from the host system to your workstation as a binary file. To download the package, you can, for example, use *FTP*. Choose as the target path name any directory where you want to store the tar file temporarily and unpack it for installation.

   Using FTP, the command is, for example:

   ```
   ftp <hostname>
   ```

   You will be prompted for your user ID and password. After logging on to your z/OS system, enter:

   ```
     binary
     get ING.SINGPWS1(INGPTEC) <PATH>/satec.tar
     quit
   ```

2. At your workstation, enter:

   ```
     cd <PATH>
   ```

3. Unpack the package file `<PATH>/satec.tar`:

   ```
   tar -xvf <PATH>/satec.tar
   ```

   This will unpack the workstation code for subsequent installation into the current directory (*<PATH>*).

   On Windows NT, you can find the `tar` command in

   ```
   c:\tivoli\bin\w32-ix86\tools\tar.exe
   ```

4. Install the appropriate Tivoli install package
   a. From the Tivoli desktop select ***Install->Install Product*** and follow the Install Product dialog
   b. Set the media path to the *<PATH>* which contains the SA z/OS specific install packages.
   c. Select the product to be installed.
   d. Close the Install Product dialog after installation.

5. Verify the installation. The files listed in Table 23 should be stored in the correct directories.

**Note:**

After installation, the binary files are stored in the following directory:

```
$BINDIR/SAOS390/NotificationService
```

The environment variable ***BINDIR*** is set when installing the Tivoli Framework. As default, it points to

```
/usr/local/Tivoli/bin/$INTERP
```

The environment variable ***INTERP*** denotes the platform where Tivoli is used, and can be for example *aix4-r1*.

*Table 23. Notification Service Product Workstation Files*

| Member Name | Type | Purpose |
|---|---|---|
| tecad_sa390msg.baroc | TEC *baroc* file | Defines event classes |
| tecad_sa390msg.rls | TEC *rls* file | Defines rules |
| nvcons.ksh | Korn shell script (*ksh*) | Starts NetView 3270 Management Console for UNIX |

Table 23. Notification Service Product Workstation Files (continued)

| Member Name | Type | Purpose |
|---|---|---|
| nvcons.bat | Batch file | Starts NetView 3270 Management Console for Windows NT |
| tecad_sa390msg.tll | Task library definition | Contains the task library |

If the NetView 3270 Management Console is not yet installed on your workstation, you can download it from the Internet:

```
http://www.ibm.com/software/support/
```

## Activating the Installed Files

You need to activate files of the following type:

- rls files
- baroc files
- tll files

### Loading Classes and Rules

After downloading the files on the Tivoli workstation, several files are available in the directory

```
$BINDIR/SAOS390/NotificationService
```

The following instructions describe the steps required to activate the installed files at the TEC event server. These steps are necessary in order to exploit the Event/Automation Service to send SA z/OS events to TEC. See the *Tivoli Enterprise Console Reference Manual* for a detailed description of the following commands:

- Use an existing rule base with classes imported.
- Import the class file (.baroc) into the rule base:

  ***wrb-imprbclass tecad_sa390msg.baroc <rbname>***
- Import the rules file (.rls) into the rule base:

  ***wrb -imprbrule tecad_sa390msg.rls <rbname>***
- Compile the rule base:

  ***wrb -comprules <rbname>***
- Load the rule base into the TEC event server:

  ***wrb -loadrb -use <rbname>***
- Stop the TEC event server:

  ***wstopesvr***
- Start the TEC event server:

  ***wstartesvr***

### Creating the System Automation Task Library

If the NetView 3270 Management Console is installed on a workstation in your network managed by Tivoli, you can use a task provided in the System Automation Task Library to start the NetView Client from the Tivoli Enterprise Console.

Depending on the platform where the NetView 3270 Management Console is installed, either a shell script (`nvcons.ksh`) for UNIX platforms or a batch file (`nvcons.bat`) for the Windows NT platform needs to be modified.

**Note:**

**Windows NT setup:**
- Modify the PATH variable via ***System Setup*** and add the path where ***Java*** is installed.
- Modify ***Tivoli Object Dispatcher Service*** via ***System Setup*** and allow the service to interact with the desktop in order to display the NetView Console.

- Edit the according file as appropriate:

  **CLASSPATH variable (UNIX and Windows NT)**
  Set to the path where Java classes are installed, add NetView class directory (see NetView documentation for details).

  **NV variable (UNIX only)**
  Set to path where NetView 3270 Management Console is installed on your system.

  **Java variable (UNIX only)**
  Set to path where Java is installed on your system.

  **CD <NetViewDir> (Windows NT only)**
  Set to directory where the NetView 3270 Management Console is installed on your system.

  Example for UNIX (nvcons.ksh):

```
export NV=/IBMFLB
export JAVA=/develop_driver/java/Java
export CLASSPATH=$JAVA/classes:$NV:$NV/sguide:$NV/sguide/SGJ023A.ZIP
:$NV/sguide/sguide.zip:$NV/src/ibmflb:$NV/jhelp
```

  Example for Windows NT (nvcons.bat):

```
set CLASSPATH=.;C:/users/java/lib/classes.zip
cd \IBMFLB
```

- Import the Task Library by using the wtll command (see the *Tivoli Management Framework Reference Manual* for details about this command).

```
wtll -p <policy region> -P <preprocessor> tecad_sa390msg.tll
```

  where

  **<policy region>**
  specifies the policy region in which to create the new task library. The policy region must exist in the local TMR.

  **<preprocessor>**
  specifies the path to the program to use as a preprocessor on the import file before it is parsed. The import file tecad_sa390msg.tll does not need to be preprocessed, so instead of specifying for example a C or C++ preprocessor, the command /bin/cat could be used.

## Customization of the Tivoli Enterprise Console

To perform the steps described in this section, you should be familiar with the Tivoli terms ***event groups*** and ***event sources***. These are introduced in *Tivoli Enerprise Console User's Guide.*

In Tivoli, you may monitor events belonging to a group which may originate from a certain source or from different sources. In order to enable the TEC event server to handle the SA z/OS specific events, you may need to define the appropriate source to TEC.

To enable Tivoli administrators to monitor events on their event consoles, you need to define one or more appropriate event groups (with events from the defined event sources) and assign these groups to the respective administrators' event consoles.

Perform the following definition steps:

1. Define the *event source* NV390MSG to forward messages and NV390ALT to forward alerts to TEC
2. Define an event group by using the source and subsource attributes as filter criteria:
   - All events that originate from SA z/OS messages have SAOS390_SysOps as the subsource
   - All events that originate from SA z/OS alerts have SAOS390_ProcOps as the subsource
3. Assign the defined event group to an Tivoli administrator's Tivoli Enterprise Console

# Part 3. Appendixes

# Appendix A. Security and Authorization

This appendix describes how to install security options on your system.

## Securing Focal Point Systems and Target Systems

Your operations staff and automation facilities at both focal point system and target systems need to be authorized to manage the resources in their environment. You can control human and automation operator authority through the password security provided by either:

- NetView
  - Operator definition file (DSIOPF)
  - RODM access information
- An SAF-based security product such as RACF

NetView facilities limit the use of commands and keywords to authorized operators and limit an operator's span of control to specific systems. Access to the SA z/OS graphic interface is controlled by user ID, password, and RODM access information. SA z/OS provides the sample INGESCAT for NetView authorization.

RACF can be used to limit the use of z/OS system commands to authorized operators. SA z/OS provides the sample INGESAF for a RACF environment.

When a target system is in the same sysplex as the focal point system, and your security product supports it, it is recommended that you share security definitions.

## Granting NetView and the STC-User Access to Data Sets

This section describes what levels of access authorities you need to assign to NetView and to specific started tasks.

### Access to XCF Utilities

The CDS recovery as well as some operator commands use the XCF utilities to retrieve couple data set information. Because the DD name SYSPRINT is required by the utilities, but can also be assigned by NetView for holding log data, the call of the utilities is implemented as a started task in the PROCLIB. The input and output data sets used by the started tasks are dynamically allocated and deleted by the NetView address space. This requires the RACF ALTER access to these data sets for NetView.

When the address space of the started task is created, the operating system assigns a user ID (IBM default: STCUSER) to the started task. This user ID must have RACF UPDATE access to the data sets. The data set names are created as follows:

```
hlq.domain.HSAyyddd.Xhhmmss
```

where:

**hlq** is the high-level qualifier for temporary data set defined during the customization

**domain** is the domain ID of the current NetView

**X** is I, O, or P

## Access to HOM Interface

Sometimes after an IPL an operating system does not know its sender paths to the coupling facilities in the sysplex. In this case the automation functions call the HCD HOM interface to determine the missing path information. As the HOM interface must not run authorized the interface is called via a started task. The input and output data sets used by the started tasks are dynamically allocated and deleted by the NetView address space. This requires the RACF ALTER access to these data sets for NetView.

When the address space of the started task is created, the operating system assigns a user ID (IBM default: STCUSER) to the started task. This user ID must have RACF UPDATE access to the data sets. The data set names are created as follows:

```
hlq.domain.HSAyyddd.Xhhmmss
```

where:

**hlq** is the high-level qualifier for temporary data set defined during the customization

**domain** is the domain ID of the current NetView

**X** O or P

## Access to IPL Information

The new automation function collecting, displaying, comparing, and deleting IPL information uses two started tasks. It is recommended to run the first started task immediately after an IPL as part of COMMNDxx list processing, to collect the IPL information in the SA z/OS VSAM data set ″IPLDATA″. The remaining functions are handled by a NetView command. Because the started task as well as the command can delete IPL information both need RACF CONTROL access to the VSAM data set. The started task collecting the information needs RACF READ access to all parmlib members.

When a comparison of IPL information is requested the NetView command schedules the second started task to call ISRSUPC—the compare utility provided by ISPF—as this utility requires fixed ddname. The input and output data sets used by the second started tasks are dynamically allocated and deleted by the NetView address space. This requires the RACF ALTER access to these data sets for NetView.

When the address space of the started task is created, the operating system assigns a user ID (IBM default: STCUSER) to the started task. This user ID must have RACF UPDATE access to the data sets. The data set names are created as follows:

```
hlq.domain.opid.INGPIPLx
```

where:

**hlq** is the high-level qualifier for temporary data set defined during the customization

**domain** is the domain ID of the current NetView

**opid** is the NetView operator ID

**x** L, N, or O

## Access to Spare Couple Data Sets

Because the CDS recovery allocates and deletes spare couple data sets via an XCF utility the user ID assigned to the started task address space must also have RACF ALTER access to these couple data sets. The names of the spare couple data set are built as follows:

```
hlq.cdstype.CDSnn
```

where:

**hlq** is the high-level qualifier for couple data sets defined during the customization

**cdstype** is ARM, CFRM, LOGR, SFM, SYSPLEX

**nn** is the sequence number corresponding to the volume entry in the list of volumes

## Access to User-Defined Couple Data Sets

In addition, the user ID of the started task address space needs RACF READ access to all user-defined couple data sets. And, when LOGGER recovery is enabled, the user ID needs RACF ALTER access to the LOGR couple data sets as well.

## Access to Spare Local Page Data Sets

The new auxiliary shortage recovery allocates and formats spare page data sets. For this reason NetView requires RACF ALTER access to these page data sets. The names of the spare page data set are built as follows:

```
hlq.sysname.Vvolume.Snn
```

where:

**hlq** is the high-level qualifier for page data sets defined during the customization

**sysname** is the name of system for which the data set is allocated

**volume** is the serial number of the volume on which the data set is allocated

**nn** is a unique sequence number

# Restricting Access to INGPLEX and INGCF Functions

This section describes how you can grant and control access of users to the INGCF and INGPLEX commands.

Access to sensitive functions of the INGPLEX and INGCF commands should be granted to certain operators only. To do this:

- Restrict access to the INGRCCHK command for the keyword INGPLEX or INGCF, and certain given values
- Permit certain operators or groups of operators to access these restricted commands, keywords, and values

To achieve this, use the NetView command authorization table or SAF command authorization.

The following keywords and values are applicable for restricting access to the functions of the INGPLEX and INGCF commands:

Keyword INGPLEX with value CDS allows for:

- Allocating an alternate CDS via the INGPLEX CDS command
- Controlling the SDUMP options and the SLIP traps sysplexwide

Keyword INGCF with value STR allows for:

- Forcing the deallocation of a CF structure via the INGCF STRUCTURE command
- Rebuilding a CF structure on another CF via the INGCF STRUCTURE command
- Controlling the SDUMP options and the SLIP traps sysplexwide

Keyword INGCF with value CF allows for:

- Preparing a CF for removal from the sysplex via the INGCF DRAIN command
- Integrating, or reintegrating, a CF into a sysplex via the INGCF ENABLE command
- Including keyword INGCF with value STR

Keyword INGPLEX with value HW allows for:

- Deactivating the LPAR of a CF via the INGCF DRAIN command
- Activating the LPAR of a CF (equivalent to starting the Coupling Facility Control Code) via the INGCF ENABLE command
- Including keyword INGCF with value CF

To activate the authorization check via the NetView command authorization table, add the protect and permit statements for the INGRCCHK command, the INGPLEX and INGCF keywords and the CDS, STR, CF and HW values as shown in the following example:

```
PROTECT  *.*.INGRCCHK.INGPLEX.CDS
PROTECT  *.*.INGRCCHK.INGCF.STR
PROTECT  *.*.INGRCCHK.INGCF.CF
PROTECT  *.*.INGRCCHK.INGPLEX.HW
PERMIT GRP3   *.*.INGRCCHK.INGPLEX.CDS
PERMIT GRP5  *.*.INGRCCHK.INGPLEX.HW
PERMIT GRP3  *.*.INGRCCHK.INGCF.STR
PERMIT GRP4  *.*.INGRCCHK.INGCF.CF
```

With these definitions operators of group GRP3 are authorized to issue all functions of the INGPLEX CDS and the INGCF STRUCTURE commands.

Operators of group GRP4 are authorized to issue all functions of the INGCF CF and the INGCF STRUCTURE commands, but are not authorized for the functions of the INGPLEX CDS commands.

## Security for IBM Tivoli Monitoring Products

This section describes security options for controlling access to IBM Tivoli Monitoring products (in particular for OMEGAMON XE) and to OMEGAMON classic monitors.

### Controlling Access to IBM Tivoli Monitoring Products

The IBM Tivoli Monitoring (ITM) platform offers a series of Simple Object Access Protocol (SOAP) requests that can be issued from z/OS. SOAP is a communications XML-based protocol that lets applications exchange information through the Internet. For further information about creating SOAP messages, refer to “Appendix C. Tivoli Enterprise Monitoring Web services” in *IBM Tivoli Monitoring: Administrator’s Guide*.

Authentication of users (autotasks or operators) is done based on <userid> and <password> tags that are specified in a SOAP request, if security is enabled. Note, however, that before a SOAP request can be issued the user must be logged on to NetView.

The SOAP request is sent to the hub Tivoli Enterprise Monitoring Server (monitoring server) that is supplied in the INGOMX command and processed there.

SOAP requests can be authorized in terms of both user and hub monitoring server via a user access list. They can be further restricted to groups of users and particular SOAP servers using command authorization table identifiers however final authorization is performed on the hub monitoring server based on the user access list and logon validation.

The relevant keywords that are supported by the INGOMX command are SERVER and IPADDR:

- SERVER allows access based on either the server object that is defined in the SOAP SERVER policy item of a NTW policy object, or a host name. Note that you can only specify the first 8 characters for long host names.
- IPADDR allows access based on IP addresses, however this must be for all IP addresses or none because an address cannot be specified in the command authorization table.

Table 24 on page 176 shows the SA z/OS command names, keywords, and values that can be protected along with their associated SAF resource or command authorization table identifier.

## Controlling Access to OMEGAMON Monitors

OMEGAMON provides both product level security and command level security:

- Product level security is applied when users log on to OMEGAMON
- Command level security is applied when users issue commands

A generic SA z/OS user ID must be defined to SAF for external product level security or to OMEGAMON for internal product level security.

For commands that are protected only by internal security, command locking must be enabled for this user ID, based on the command authority level needed by SA z/OS. For example, if only level 0 and 1 commands are issued from SA z/OS, an INITIAL1 rule must be defined and permission must be granted to the generic user, and at the same time there must be no INITIALƀ rule. In the absence of INITIAL*n* rules, the command authority level for SA z/OS is always 0. For further details, refer to the OMEGAMON documentation.

For commands protected by external security, appropriate command resource profiles have to be created and permission must be granted to the generic user.

Note that even though the SA z/OS generic user has the potential to issue any level *n* command, you can use NetView command security to selectively define (on an operator by operator or group by group basis) which operator or group can issue a particular command.

## NetView Command Authorization

Because SA z/OS uses a common user ID that establishes sessions between SA z/OS and any OMEGAMON, SA z/OS uses NetView and the command authorization table to control access to:

- OMEGAMON sessions
- OMEGAMON commands
- The administration of OMEGAMON sessions

For details about the command authorization table, refer to the *NetView Security Reference* manual.

The common user ID that is specified with the OMEGAMON session definitions represents the set of users (autotasks, operators) that interact with OMEGAMON sessions. It needs to be defined to OMEGAMON with the highest security level that has been granted to automation. This approach simplifies the customization that is required in OMEGAMON to permit access to the monitor.

Table 24 shows the new SA z/OS command names, keywords, and values that can be protected along with their associated SAF resource or command authorization table identifier.

*Table 24. Command Authorization Identifiers*

| Commands and Keywords | Command List Name | SAF Resource or Command Authorization Table Identifier |
|---|---|---|
| INGOMX<br>NAME<br>CMD<br>SERVER<br>IPADDR | INGROMX0 | *netid.luname*.INGROMX0<br>*netid.luname*.INGROMX0.NAME.*session_name*<br>*netid.luname*.INGROMX0.CMD.*command*<br>*netid.luname*.INGROMX0.SERVER.*server_name*<br>*netid.luname*.INGROMX0.IPADDR |
| INGSESS<br>REQ<br>START<br>STOP | INGRYSS0 | *netid.luname*.INGRYSS0<br>*netid.luname*.INGRYSS0.REQ<br>*netid.luname*.INGRYSS0.REQ.START<br>*netid.luname*.INGRYSS0.REQ.STOP |

**Notes**

1. For OMEGAMON commands that contain a period, replace it with an '@' when defining the command authorization entry, for example, to protect `.RMF` use:

   ```
   PROTECT *.*.INGROMX0.CMD.@RMF
   ```

2. If you want to use TRAP for OMEGAMON for IMS, CMD authorization for XIMS must be given and for the other monitors, CMD authorization for EXSY must be given.

Consider adopting the following approach to defining command authorization:

- For maximum security, protect all sessions and all commands.
- Permit access to sessions and commands only as needed.
- Administrators need INGOMX-NAME and INGSESS-REQ authorization.

## Password Management

Logging on to OMEGAMON requires authentication with a user ID and password if product level security is active. Note that when a password is specified, it appears in readable format in the automation configuration file and in logs. When SAFPWD is specified, the password is stored in a VSAM data set in an encrypted format.

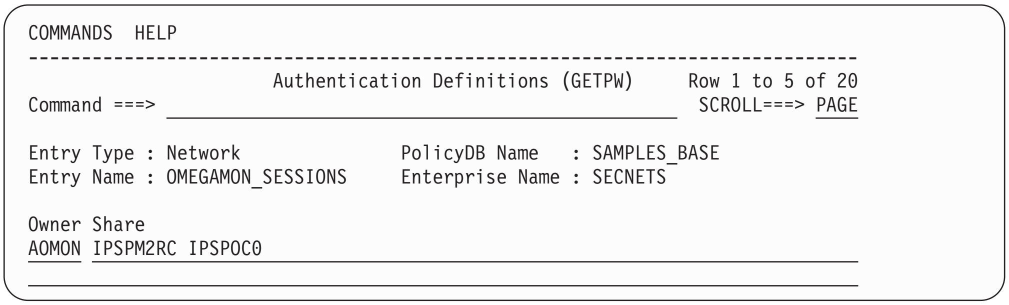
The NetView command GETPW is used to access the password data set to set or read the password.

SA z/OS uses GETPW as follows:

- Passwords are stored and retrieved by *userid* and *owner*
- *userid* is the common user defined to log on to an OMEGAMON session
- *owner* is a custom value representing one or more VTAM application IDs as defined in the authentication policy
- If no owner is defined for an application ID, it defaults to the 5 leftmost characters of the application ID

To use SAFPWD, an authentication policy item in the network policy has to be specified where all applications denoted by the OMEGAMON applid that share the same password are assigned to a single owner.

To define the authentication policy select the AUTHENTICATION policy item for the appropriate network policy. In the Authentication Definitions panel enter your definitions in the Owner and Share fields, as shown in Figure 26.



```
COMMANDS  HELP
-------------------------------------------------------------------------------
                       Authentication Definitions (GETPW)     Row 1 to 5 of 20
Command ===>                                                   SCROLL===> PAGE

Entry Type : Network              PolicyDB Name   : SAMPLES_BASE
Entry Name : OMEGAMON_SESSIONS    Enterprise Name : SECNETS

Owner Share
AOMON IPSPM2RC IPSPOC0
```

*Figure 26. Authentication Definitions Panel for Sessions*

Where the fields have the following meanings:

**Owner**
Custom value, up to 5 characters, used by SA z/OS to look up the password in the NetView password data set when an OMEGAMON session with any of the VTAM application IDs listed next is started.

**Share**
List of VTAM application IDs representing OMEGAMON monitors for which passwords are kept in the NetView password data set under the same owner key.

**Authentication Using the NetView Password Data Set:** The NetView password data set is used as a password safe if you do not want to reveal passwords in your policy database. The password data set has to be created first and allocated upon the start of NetView. Refer to *NetView Installation: Configuring Additional Components* for details.

You are responsible for setting the initial password for a user ID with a given owner in the password data set using the NetView command GETPW. Whenever a logon is made to OMEGAMON, for sessions with SAFPW defined as the user password, SA z/OS attempts to look up that user's password in the password data set. If the lookup succeeds, GETPW returns either the current password or, if the 30-day validity period has expired, the current and a new password.

On logging on to OMEGAMON, the current password is used to authenticate the user ID. If a new password is available, the new password is also changed on the OMEGAMON logon screen. Upon successful password update in OMEGAMON, the new password is also updated in the password data set using GETPW.

You are responsible for ensuring that the password in the password data set and the password known to SAF or OMEGAMON are the same, in particular when shared SAF databases are used in a multisystem complex, for example, a Parallel Sysplex. In this case, the password data sets should also be shared by the same group of systems.

Use the GETPW command to initialize the password data set. For example, suppose the session and password share definitions are set as in Figure 26 on page 177 for user `oper1` and owner `AOMON`, the GETPW command format would be:

```
GETPW oper1 AOMON,INIT=pw,MASK=@(#) 82 1.30@(#)N%N@(#) 82 1.30@(#)A@(#) 82 1.30@(#)A%A
```

where *pw* is the initial password for the user ID and the MASK parameter indicates that the password should be 8 characters long, beginning with a letter, followed by 2 numbers and then 5 letters.

See *Tivoli NetView for z/OS Command Reference Volume 1* for further details about the GETPW command.

## Controlling Access to the Processor Hardware Functions

For processor operations SNMP processor connections and for the Parallel Sysplex enhancements functions that use the BCP internal interface, a SAF product such as RACF must be used to define the required resources and grant access to these resources for the authorized NetView users and autotasks.

### Allowing NetView to Use the BCP Internal Interface

Before you can use the enhanced sysplex functions of SA z/OS for CF or XCF automation, the hardware resource (HSAET32) must be defined in NetView.

1. Define resource HSA.ET32OAN.HSAET32 in the CLASS FACILITY
2. Permit NetView READ ACCESS to this facility class resource

The following example shows the RACF commands used to define the resource and to grant the required READ access for the NetView user.

```
SETROPTS CLASSACT(FACILITY)
SETROPTS RACLIST(FACILITY)
RDEFINE FACILITY HSA.ET32OAN.HSAET32 UACC(NONE)
PERMIT HSA.ET32OAN.HSAET32 CLASS(FACILITY) ID(stcuser) ACC(READ)
```

With the SETROPTS command, the RACF class FACILITY is made available. With the SETROPTS RACLIST command the FACILITY class resource profile copy in the RACF data space is enabled to increase performance. The next command, RDEFINE, fully qualifies the HSAET32 resource and sets universal access to none. With the PERMIT command, the RACF defined user *stcuser* gets READ access to this resource. User ID *stcuser* must be the user ID associated with your NetView started task. If you start NetView as a regular job, the user ID submitting the job must be authorized for the resource.

Note that you can use a wildcard character to specify the resource more generic if that is suitable for your environment.

## Access to the CPCs

Each processor (CPC) defined in your SA z/OS policy data base must have a corresponding resource profile defined with your SAF product. Note that this only applies for processors defined with a connection type SNMP or INTERNAL.

The skeleton of the CPC resource is:

```
HSA.ET32TGT.netid.nau
HSA.ET32TGT.netid.nau.lpar
```

The netid.nau part of the resource name corresponds with the netid.nau definition of the CPC entry specified in the customization dialog. The period between netid and nau is part of the resource name. For LPAR protection define a resource with the netid.nau.lpar specification.

The following example shows how to define a CPC resource in RACF.

```
RDEFINE FACILITY HSA.ET32TGT.DEIBMD1.X7F1F30A UACC(NONE)
```

The CPC with netid DEIBMD1 and nau X7F1F30A is defined as a resource in the RACF class facility with a universal access attribute of NONE.

Note that you can use a wildcard character to specify the resource more generic if that is suitable for your environment.

## Levels of CPC Access

The following lists the access levels and their meaning for the CPC resources.

- READ—Retrieve, get configuration information from the CPC
- WRITE—Update, set configuration information of the CPC
- CONTROL—Issue operations management commands of the CPC

Note: this access level scheme is for the CPC and its LPARs.

## Defining the CPC Access Lists

Depending on the NetView operator security chosen, the access level is checked differently. If your NetView operator security (OPERSEC) is set to MINIMAL, NETVPW, or SAFPW, the user ID that is checked for hardware access is always the user ID that started the NetView address space, which is usually a STC user ID. This user ID has to be authorized for all CPC and CPC.Lpar resources you want to manage with this NetView. If multiple users are allowed to start NetView, make sure they are all authorized.

If you have chosen a NetView operator security level of OPERSEC=SAFDEF or OPERSEC=SAFCHECK, the following paragraph applies.

With SA z/OS, several NetView autotasks need to be authorized to access the CPCs that are defined in the customization dialog.

The following NetView autotasks need to be authorized with access level CONTROL for **all** defined CPCs and all its LPARs:

- The XCF and RPC autotasks
- The autotasks defined with SYN %AOFOPXCFOPER% and %AOFOPRPCOPER% in automation table member AOFMSGSY
- The HW interface autotasks AUTHW*xxx*
- Any operator issuing a HW action with INGCF

The AUTXCFxx autotasks plus the additional ones from %AOFOPXCFOPER% are used internally once INGCF drain or INGCF enable is invoked by an authorized user. IXC102A message automation is also performed by these autotasks.

The autotasks used for the HW interface initialization and communication also need to be authorized. Use access level CONTROL for the AUTHWxxx autotasks in your environment.

The following example shows how to permit access to a CPC resource in RACF:

```
PERMIT HSA.ET32TGT.DEIBMD1.X7F1F30A CLASS(FACILITY) ID(AUTXCF) ACC(CONTROL)
```

The XCF autotask AUTXCF gets access level CONTROL for the CPC resource DEIBMD1.X7F1F30A.

LPAR access example:

```
PERMIT HSA.ET32TGT.DEIBMD1.X7F1F30A.* CLASS(FACILITY) ID(AUTXCF) ACC(CONTROL)
```

The XCF autotask AUTXCF gets access level CONTROL for the CPC resource DEIBMD1.X7F1F30A and all its defined logical partitions.

### Implementing Granular Hardware Access

By giving operators READ access to a CPC resource and CONTROL access only to LPARS according to the business needs, a flexible security scheme can be implemented.

## Defining an RACF Profile for I/O Operations

Assign authorization levels using RACF/SAF for individual commands or generically for all commands. Use the RACF RDEF command with a class of FACILITY.

| Function | Command |
|---|---|
| To define the profile for the PROHIBIT command | RDEF FACILITY IHV.PROHIBIT |
| To define a profile that would allow all users to enter a command (for example, UNLOCK) | RDEF FACILITY IHV.UNLOCK UACC(READ) |
| To permit the use of generics for a Class of Service facility | SETROPTS GENERIC FACILITY |
| To prevent unauthorized use of commands you can enter this RACF command to prohibit use of commands | RDEF FACILITY IHV.* UACC(NONE) |

**Note:** If you have prohibited all user IDs from using these commands, you must explicitly assign RACF authorization to designated user IDs.

## Assign RACF Authorization

To give RACF authorization to a user ID, enter the RACF PERMIT command and its parameters.

### Assign a Profile Parameter

The profile parameter is `IHV`*`commandname`*, where:

- `IHV.` is the three-character ID, followed by a period (.).
- *commandname* is the name of the command

**Notes:**

1. The profile parameter (for example, IHV.ALLOW, IHV.VARY, IHV.REMOVE.SWITCH) determines the authorization level of the user ID identified in the ID parameter.
2. The ACCESS parameter identifies the authorization given.

   You can use an asterisk to designate a generic class on the PERMIT parameters. For example, to allow all users to send all commands that require read authority, enter:

   ```
   PERMIT IHV.* ACCESS(READ) CLASS(FACILITY)
   ID(*)
   ```

## Assign Authorization by ACCESS Level

You can authorize a user ID to enter one command at a given access level by entering one command.

For example, to allow a user (SUWAJDA) to send commands requiring control authorization, enter:

```
PERMIT IHV.* ACCESS(CONTROL) CLASS(FACILITY)
ID(SUWAJDA)
```

For example, to authorize another user (FISHER) to enter all commands that require the update authorization, enter:

```
PERMIT IHV.* ACCESS(UPDATE) CLASS(FACILITY)
ID(FISHER)
```

### Assign Authorization by Command

You can use the PERMIT command to let all users send individual commands. For example, to authorize everyone to use the Unlock command, enter:

```
PERMIT IHV.UNLOCK ACCESS(READ) CLASS(FACILITY)
ID(*)
```

To authorize a user (DONC) to send all connectivity commands with the Noforce option, enter:

```
PERMIT IHV.* ACCESS(UPDATE) CLASS(FACILITY)
ID(DONC)
```

### Use Specific Profile Names

Either specific profile names or generic profile names can be used in the PERMIT command. Use specific profile names to authorize use of specific I/O operations commands.

For example, to authorize a user (PHILOP) to use only the Allow and Prohibit commands with the Noforce option, enter:

```
PERMIT ING.ALLOW ACCESS(UPDATE) CLASS(FACILITY) ID(PHILOP)
PERMIT ING.PROHIBIT ACCESS(UPDATE) CLASS(FACILITY) ID(PHILOP)
```

On the NMC focal point the following is necessary to define users and access levels to RODM:

1. Define a general resources class named RODMMGR. This is the default class name used in EKGCUST initialization member for RODM.
2. Define instances of the RODMMGR resource class, for example,

```
RDEF EKGXRODM1 CLASS(RODMMGR) UACC(NONE)
RDEF EKGXRODM2 CLASS(RODMMGR) UACC(NONE)
RDEF EKGXRODM3 CLASS(RODMMGR) UACC(NONE)
RDEF EKGXRODM4 CLASS(RODMMGR) UACC(NONE)
RDEF EKGXRODM5 CLASS(RODMMGR) UACC(NONE)
RDEF EKGXRODM6 CLASS(RODMMGR) UACC(NONE)
```

For more information on the RACF commands, see *Resource Access Control Facility (RACF) Command Language Reference*.

### Assign TCP/IP Port Authorization

When the physical file system is configured as INET, RACF can be used to restrict access to the ports that are used by I/O operations when using TCP/IP communication. For details about how to restrict access, refer to the section “Port access control” in “Chapter 3. Security” of *z/OS Communications Server: IP Configuration Guide*.

## Establishing Authorization with Network Security Program

If you have installed Network Security Program (NetSP), you can create an authorization system requiring only one sign on for each user. With it, a user who logs on from a workstation has access to RACF-protected host applications. These include 3270 emulation and log on scripts and APPC communications. This authorization is controlled by NetSP’s PassTicket, which is recognized by the SAF-based security system and is valid for a fixed period of time.

To establish authorization for your users, you need to create in NetSP recorded input files as log on transfer scripts. This is done either by recording keystrokes in the emulator session or by entering them directly in a file with a text editor. How to do this is described in *Network Security Product Secured Network Gateway Guide*.

# Appendix B. Syntax for INGTOPOF File

The INGTOPOF file contains configuration information for the SA z/OS topology manager. It must reside in any of the data sets allocated under the DSIPARM concatenation. The records of the file consist of a keyword with one or more parameters. Comment lines must start with an asterisk (*). A '+' at the end of a line indicates that the record is continued in the next line.

The following keywords can occur in the INGTOPOF file: SYSPLEX, PROCOPS, LOCATION, ANCHOR, BLDVIEWS, OPTION, and TEMPLATE.

## The SYSPLEX Statement

For every sysplex, the SA z/OS topology manager must be told which systems of the sysplex are able to communicate with it. This is done with the SYSPLEX statement according to the following format:

```
                               ┌──────────────────────────┐
                               ▼                          │
►►──SYSPLEX──sysplex_name──┬──────────┬──domain_id──┴─────────────────────►◄
                           └─net_id.──┘
```

The *sysplex_name* must be different from every name that you specify in a PROCOPS statement (see "The PROCOPS Statement" on page 184). The systems must be identified to the SA z/OS topology manager by their NetView domain ID. If the *net_id* is omitted, it is assumed to be the same as that of the focal point. The INGTOPOF file must contain at least one SYSPLEX statement; in particular, you cannot have a PROCOPS statement in the INGTOPOF file without a SYSPLEX statement.

The SA z/OS topology manager tries to contact the systems in the order in which they appear in the list. When it finds a system that contains a functional SA z/OS topology agent, it searches no further, but gathers the SA z/OS information from the automation manager through this SA z/OS topology agent. It then stores the retrieved information in RODM, prefixing all resource names with the *sysplex_name* that it found in the SYSPLEX statement.

It follows from this that the order in which the domains are specified should reflect eventual decisions about primary and backup systems for communication with the SA z/OS topology manager. Also, the sysplexes as defined in the INGTOPOF file must correspond to the sysplex groups in the policy database.

Because standalone systems are treated as sysplexes, they must also be introduced to the SA z/OS topology manager by a SYSPLEX statement. In this case, the list of domain IDs will comprise just one item.

If you want to have a network anchor for a system, this system's domain ID must be included in the SYSPLEX statement.

## The PROCOPS Statement

With this statement, you specify a focal point for processor operations and its backup focal point. It has the following format:

```
►►──PROCOPS──procops_name──focal_point──backup_focal_point──────────────────────►◄
```

The *procops_name* must be different from every name that you specify in a SYSPLEX statement. The focal point processor and its backup must be identified to the SA z/OS topology manager by a NetView domain ID. If the *net_id* is omitted, the SA z/OS topology manager assumes it to be identical to that of its own focal point.

There must be at least one SYSPLEX statement in the INGTOPOF file if you want to insert a PROCOPS statement.

## The LOCATION Statement

The LOCATION statement is used to group system related events, for example, geographically rather than logically. The events that are attached to a LOCATION must be posted to the SA z/OS topology manager by the user with the INGPOST command. For more information on the INGPOST command, see *IBM Tivoli System Automation for z/OS Operator's Commands*.

The Location statement has the following format:

```
►►──LOCATION──target_domain──location_name──────────────────────────────────────►◄
```

***Examples:***

```
*
* TSCF1 thru 3 are in Boeblingen, 4 and 5 are in Perth
*
LOCATION T2 BB_LAB
LOCATION NETOZ.CNMT4 PERTH
LOCATION NETOZ.CNMT5 PERTH
*
* AOCA thru D are in Boeblingen
*
LOCATION AOCPLEX BB_LAB
*
* OZ1 thru OZ4 are in Perth
*
LOCATION OZPLEX PERTH
```

## The ANCHOR Statement

ANCHORS are entered via the customization dialogs on the target systems. For more information about how to define anchors see *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

The ANCHOR statement will remain in the INGTOPOF to allow ANCHORs to be defined for downlevel systems where ANCHORS are not entered via the customization dialogs.

ANCHORs for downlevel systems will occur in RODM, but not in the automation manager.

The ANCHOR statement serves to define anchors for arbitrary user defined events.

Anchors serve to collect events of a certain type that are to be displayed on the NMC. Anchors play the role of major resources for events of this type, and the events themselves are treated as minor resources of their anchor. The SA z/OS topology manager automatically creates anchors for heartbeats but not for WTOR or tape mount requests.For more information on anchors and events see *IBM Tivoli System Automation for z/OS User's Guide*.

With the ANCHOR statement, you can introduce your own anchors for any events. These events must be posted to the SA z/OS topology manager with the INGPOST command; the anchor must be specified in the command as the major resource (RESOURCE parameter). For more information on the INGPOST command, see *IBM Tivoli System Automation for z/OS Operator's Commands*; for information on major and minor resources, see *IBM Tivoli System Automation for z/OS Defining Automation Policy*

## The BLDVIEWS Statement

A RODM resource can only be displayed on the NMC when it is included in a view. With the BLDVIEWS statement, you can pass data sets (members) that contain view definitions for BLDVIEWS to the SA z/OS topology manager. The SA z/OS topology manager will then call the BLDVIEWS tool for (all or some of) these data sets (members) in order to build or rebuild the specified views. The view definitions must be supplied by the installation.

Every BLDVIEWS statement associates one sysplex (as defined by a SYSPLEX statement) or one processor operations focal point configuration (as defined by a PROCOPS statement) with a list of such data sets (members). This enables the SA z/OS topology manager to rebuild views at runtime only for those sysplexes (sets of target processors) whose SA z/OS information has in fact changed.

The BLDVIEWS statement has the following format:

```
                                  ┌─────────────────────────┐
                                  ▼                         │
►►──BLDVIEWS──┬─sysplex_name─┬──────data_set_or_member──────┴──────────────────────►◄
              └─procops_name─┘
```

You can exploit the association of the data sets (members) to sysplexes to reduce the overhead caused by rebuilding views at runtime. Suppose, for example, that all your sysplex views either contain objects from only one sysplex or from all sysplexes. Then you should proceed as follows.

1. For every sysplex, create a separate data set (member) with the view definitions specific for that sysplex.
2. Create one data set (member) for the common views.
3. Code a BLDVIEWS statement for every sysplex, where the list of data sets (members) comprises two items, namely the data set (member) with the views specific for this sysplex, and the data set (member) with the common views.

In this way, the sysplex specific views are rebuilt only when the SA z/OS resources for the sysplex in question have changed in RODM in such a way that a rebuild is necessary.

For more details on view definitions, see *IBM Tivoli System Automation for z/OS User's Guide*.

## The OPTION Statement

With the OPTION statements you can:

- control whether or not dependencies and major/minor resource relationships are stored in RODM, and are therefore represented on the NMC, and
- specify the default layout for the automatically generated subviews of group objects.

A separate OPTION statement is required for each option.

The OPTION statement has the following format:

```
           ┌─PARENTS───┐
►►──OPTION─┤           ├──────────────────────────────────────────────────►◄
           └─NOPARENTS─┘
```

```
           ┌─LINKMINOR───┐
►►──OPTION─┤             ├────────────────────────────────────────────────►◄
           └─NOLINKMINOR─┘
```

```
           ┌─MDL_RADLINK─┐
►►──OPTION─┤             ├────────────────────────────────────────────────►◄
           └─MDL_GRID────┘
```

The parameters have the following meaning:

**PARENTS**
Dependency relationships are stored in RODM (and displayed on the NMC in network views). This is the default.

**NOPARENTS**
Dependency relationships are not stored in RODM.

**LINKMINOR**
Relationships between major and minor resources are stored in RODM (and displayed in network views). This is the default.

**NOLINKMINOR**
Relationships between major and minor resources are not stored in RODM.

**MDL_RADLINK**
The automatically created subviews are radially arranged. This is the default. The default for this option is defined in RODM.

**MDI_GRID**
The automatically created subviews are arranged in a grid. The default for this option is defined in RODM.

If you want to use the default values, no explicit OPTION statement is required.

## The TEMPLATE Statement

The name displayed beneath a resource on the NMC is the DisplayResourceName field of the resource. It can be customized using the TEMPLATE parameter in the INGTOPOF file. The template entries in the INGTOPOF file control how the DisplayResourceName of a resource is formatted.

When using the locate function on the NMC, it is the DisplayResourceName field of the resource that is compared with the search criteria of the locate for an exact match.

It is not a requirement to have any template parameters in the INGTOPOF file. If no template parameter is found in INGTOPOF, the format of the DisplayResourceName will default to the following:

- `PLEX.SYSTEM.TYPE.SUBSYSTEM EVENT` for major resources
- `PLEX.SYSTEM.TYPE.SUBSYSTEM.MINOR EVENT` for minor resources

To change the format of the default DisplayResourceName, special *type* templates are required to specify how the default DisplayResourceName should be formatted. There are the following two types:

- DRN for major resources
- DRNM for minor resources

Customization of the DisplayResourceName can be defined for all resource types (DRN, DRNM), or individually for each resource type (APL, APLM, APG, APGM).

When a resource is created, the type (for example, APL or APG) of the resource is searched for in the INGTOPOF file to find a matching template.

- If a match is found, the DisplayResourceName will be formatted as specified by the type template in the INGTOPOF file.
- If no match is found, the DisplayResourceName will be formatted using the default.

The major resource types supported by the template parameter in the INGTOPOF file are:

| | |
|---|---|
| **APL** | applications |
| **APG** | application groups |
| **APGP** | application groups (sysplex) |
| **SYS** | system |
| **SYG** | system groups |
| **GRP** | groups |
| **MTR** | monitor resources |

Because minor resources can be attached to major resources, the following types are also supported by the template parameter in the INGTOPOF file for minor resources:

| | |
|---|---|
| **APLM** | application minors |
| **APGM** | application group minors |
| **APGPM** | application group (sysplex) minors |

| | |
|---|---|
| **SYSM** | system minors |
| **SYGM** | system group minors |
| **GRPM** | group minors |
| **HEARTBEATM** | heartbeat minors |
| **WTORM** | WTOR minors |
| **TAPEM** | tape minors |
| **CFM** | coupling facility minors |
| **CDSM** | coupled data set minors |
| **ETRM** | external timer minors |
| **SYSPLEXM** | sysplex minors |
| **MTRM** | monitor resource minors |

All, any, or none of the above type templates can be used.

When user defined anchors are created, the following applies:

- If the format of the default DisplayResourceName is acceptable, no additional template will be required in the INGTOPOF file. The DisplayResourceName is formatted using the default.
- If you customized the format of the DisplayResourceName, it is necessary to create a template for the user-defined anchors, to specify how the DisplayResourceName must be formatted for the user-defined anchors and any minor resources attached to the user-defined anchors.

If the anchor statement `ANCHOR K1 USER` exists in the INGTOPOF file, define the following two type templates in the INGTOPOF file to control the formatting of the DisplayResourceName for the anchor and any attached minor resources:

- USER for the anchor
- USERM for the minor resources attached to the anchor

To define how the DisplayResourceName is formatted, substitution parameters are employed. Substitution parameters can appear in any order. The following substitution parameters are supported:

| | |
|---|---|
| **&STR.** | system.type.subsystem |
| **&RES.** | subsystem/type/system |
| **&MNR.** | minor resource name (minor resources only) |
| **&SUB.** | subsystem |
| **&TYP.** | type |
| **&SYS.** | system |
| **&EVT.** | event |
| **&PLX.** | sysplex |
| **&DATE.** | date |
| **&TIME.** | time |

If event (&EVT.) is specified as a substitution parameter and no event field exists for the resource, an * is inserted in the DisplayResourceName. If a substitution field does not exist for a resource where a substitution parameter has been specified, the substitution parameter itself (for example, &SYS. &STR.) will appear in its place in the DisplayResourceName.

## Examples

**Customizing DisplayResourceName for APLs:**

If the requested DisplayResourceName for APLs was system name and subsystem name (for example, SYSX.RODMX), the following entry would be required in the INGTOPOF file:

```
TEMPLATE APL &SYS..&SUB.
```

**Customizing DisplayResourceName for all resources:**

If the requested DisplayResourceName for all resources was system name, subsystem name, and event (for example, SYSX.RODMX Event Text), the following entry would be required in the INGTOPOF file:

```
TEMPLATE DRN &SYS..&SUB. &EVT.
```

**Customizing DisplayResourceName for user anchors:**

The following anchor statement is found in INGTOPOF file:

```
ANCHOR K1 PLEX1
```

If the requested DisplayResourceName was subsystem, date, and time (for example, RODMX 19 MAY 2002.02:16:45), the following entry would be required in the INGTOPOF file:

```
TEMPLATE PLEX1 &SUB. &DATE..&TIME.
```

The above examples are for major resources. If customization of the DisplayResourceName is also required for minor resources attached to the major resources, similar template entries in the INGTOPOF file would be required:

- TEMPLATE APLM &SYS..&SUB..&MNR.
- TEMPLATE DRNM &SYS..&SUB..&MNR. &EVT.
- TEMPLATE PLEX1M &SUB..&MNR. &DATE..&TIME.

For an example of the template statements in the INGTOPOF file, refer to “Sample INGTOPOF File” on page 192.

As the DisplayResourceName can now be customized, it is possible to create different resources with the same DisplayResourceName. Although duplicate DisplayResourceNames cause no problems to the NMC or RODM, it will be the responsibility of each installation to ensure that any duplication is correctly processed by any user-written code.

BLDVIEWS creates views containing resources, and can identify resources for inclusion by the MyName field or the DisplayResourceName field of the resource.

- No further change to your BLDVIEWS statements will be required.
- The format of the MyName field may NOT be modified.
- The format of the MyName field is, `PLEX.SUBSYSTEM/TYPE/SYSTEM.MINOR`
- The MyName field may have parts omitted that are not relevant.

- The following are examples of the MyName:

  ```
  PLEX.SUBSYSTEM/TYPE              - major
  PLEX.SUBSYSTEM/TYPE/SYSTEM       - major
  PLEX.SUBSYSTEM/TYPE.MINOR        - minor
  PLEX.SUBSYSTEM/TYPE/SYSTEM.MINOR - minor
  ```
- If you currently use the DisplayResourceName in your BLDVIEWS statements and you are customizing the DisplayResourceName, it will be necessary to review your BLDVIEWS statements to ensure that the correct resources are included in your views.

## The RUNOPID Statement

When submitting commands via the NMC, the commands are run under the user ID of the operator signed on to the NMC at that time.

It is possible to select a predefined user ID by using the RUNOPID statement in the INGTOPOF file. When a command is submitted via the NMC for a non-local resource, the command will be run under the predefined user ID, and not the user ID of the operator signed on to the NMC at that time.

Commands that are issued via the NMC against a local resource are never preceded by a label.

Commands that are issued via the NMC against a non-local resource are preceded by a label. This label has three separate fields:

- Netid
- Domain
- User id

Examples of the label are as follows:

- `Netid:`
- `Netid.Domain:`
- `Netid.Domain/User id:`

To provide an amount of flexibility, the RUNOPID statement has been introduced to the INGTOPOF file. This will allow a predefined user ID to be used in the label, rather than the user ID of the operator signed on to the NMC at that time.

If the RUNOPID statement exists in the INGTOPOF file, the associated user ID will be substituted in the label.

The syntax of the RUNOPID statement in the INGTOPOF file is

```
RUNOPID user id
```

An example of the RUNOPID statement in the INGTOPOF file is

```
RUNOPID ACDMON
```

If multiple RUNOPID statements appear in the INGTOPOF file, only the first RUNOPID statement will be used, all subsequent RUNOPID statements will be discarded.

## The HBDELETE Statement

The HBDELETE statement specifies whether or not old heartbeat entries should be deleted. The default is yes, which provides behavior consistent with earlier releases. The syntax is:

```
►►──HBDELETE──┬─Y─┬─────────────────────────────────────────────────►◄
              └─N─┘
```

When Y is specified, all previous heartbeat minor resources from the same sysplex are deleted when any heartbeat minor resource from the sysplex is updated. This incurs a measurable resource consumption.

When N is specified, only the update to the heartbeat minor resource is made. This means that RODM may end up containing old (stopped or failed) heartbeats from other systems in the sysplex, long after the heartbeat has been picked up by another system in the sysplex. This is measurably more efficient than the Y option.

## The LINKTOVIEWS Statement

The LINKTOVIEWS statement determines which RODM fields will be used to connect major and minor resources in RODM. Specifying BASE or NONE makes processing faster, but at the cost of losing some NMC functionality. The syntax is:

```
►►──LINKTOVIEWS──resource──linkage──────────────────────────────────►◄
```

The *resource* parameter may be either a qualified major resource name (sysplex.major), a sysplex name (sysplex) or the constant 'DEFAULT'.

The *linkage* values are:

**FULL** All links are made, this is the default behavior. Fields linked are:

- IsPartOf/ComposedOfLogical
- ContainedInView
- Aggregationparent
- ExceptionViewList

.

**BASE** The only fields linked are IsPartOf/ComposedOfLogical and AggregationParent. The missing fields mean the minor resource will not appear in any views containing the major resource and will not appear in any exception views containing the major resource (unless placed there by an alternate mechanism such as RCM or BLDVIEWS).

**NONE**
No links are made, the minor resources will not be accessible from NMC unless picked up by something such as BLDVIEWS or RCM.

## The MAPCOLOR Statement

The color for a resource icon of status “Unavailable” can be changed with the keyword “MAPCOLOR”. The updated color will be displayed on all NMC topology clients. The syntax is:

```
►►──MAPCOLOR──UNAVAILABLE──┬─user positive value─┬──────────────────────────►◄
                           └─user negative value─┘
```

It is possible to map the status of “Unavailable” to all “User positive” and “User negative” values. These are:

- User positive: 136 137 138 139 140 141 142 143
- User negative: 152 153 154 155 156 157 158 159

**Example:**

The default dark green color can be changed to light green by placing the following line in the topology file (INGTOPOF):

```
MAPCOLOR UNAVAILABLE 136
```

On the NMC topology client, the color of each “User positive” or “User negative” value can be displayed and changed with:

Options ► Console properties... ► Status

**Technical Note:**

Refer to the RODM **DisplayStatus** field in *Tivoli NetView for z/OS Data Model Reference*.

**Note:**

The **DisplayStatus** field has a major impact on the decision whether an object should be placed in an exception view.

SA z/OS expects that the RODM and GMFHS defaults put the ’UserNegative’ values into exception views. ’UserPositive’ values are assumed not to appear in exception views.

## Sample INGTOPOF File

```
*********************************************************************
*
* INGTOPOF sample
*
* The sysplex_name in this example is:  K1
* The sysplex consists of the following four
* domains: IPSNM, IPSNN, IPSNO and IPSNP
*
* The KEY1VIEW and CMNVIEW members contain BLDVIEWS control cards.
* They are necessary for the SA topology manager to create 'views'
* in RODM to display SA resources.
* For more details refer to the SA User's Guide,
* Using the NetView Management Console for SA z/OS,
* Creating Views
```

```
*
* This sample also contains a user defined anchor 'USER' and
* shows the usage of the 'HBDELETE', 'LINKTOVIEWS', 'OPTION' and
* 'TEMPLATE' statements.
*
* For a description of all keywords please refer to the
* System Automation for z/OS Planning and Installation guide.
*
* Use a trailing '+' for continuation.
*
********************************************************************
*
SYSPLEX  K1 IPSNM IPSNN +
                  IPSNO +
                  IPSNP
*
BLDVIEWS K1 KEY1VIEW CMNVIEW
*
ANCHOR K1 USER
*
* HBDELETE N
*  When heartbeat minor resources for the SYSPLEX are updated via the INGPOST
*  command, heartbeat minor resouces will be created on receipt of the initial
*  INGPOST command, these heartbeat minor resources will then be updated for
*  subsequent INGPOSTs commands.
*
* HBDELETE Y
*  When heartbeat minor resources for the SYSPLEX are updated via the INGPOST
*  command, any existing heartbeat minor resources for the SYSPLEX will be deleted
*  and new heartbeat minor resources for the SYSPLEX will be created.
*
* In the following LINKTOVIEWS examples,
* o The sysplex is 'K1',
* o The major resource is 'KEY1/SYS/KEY1'
*
* LINKTOVIEWS DEFAULT FULL
* LINKTOVIEWS K1 BASE
* LINKTOVIEWS K1.KEY1/SYS/KEY1 NONE
*
* OPTION NOPARENTS
* OPTION NOLINKMINOR
OPTION MDL_RADLINK
*
*==================================================================*
* To define how the DisplayResourceName is formatted,              *
* substitution parameters are employed. Substitution               *
* parameters may appear in any order. The following                *
* substitution parameters are supported,                           *
*                                                                  *
* &STR. - SYS.TYPE.SUB                                             *
* &RES. - SUB/TYPE/SYS                                             *
* &MNR. - MINOR RESOURCE NAME (Minor Resources only)               *
* &SUB. - SUBSYSTEM                                                *
* &TYP. - TYPE                                                     *
* &SYS. - SYSTEM (NULL FOR SYSPLEX RESOURCE)                       *
* &EVT. - EVENT                                                    *
* &PLX. - SYSPLEX                                                  *
* &DATE. - DATE                                                    *
* &TIME. - TIME                                                    *
*                                                                  *
* To activate a TEMPLATE statement remove the leading asterisk from *
* the following samples.                                           *
*==================================================================*
*
*TEMPLATE DRN &PLX..&STR. &EVT.
*TEMPLATE DRNM &PLX..&STR..&MNR. &EVT.
*
```

```
*TEMPLATE APL &SYS..&SUB.
*TEMPLATE APLM &MNR.
*
*TEMPLATE APG &PLX. &SYS. &RES.
*TEMPLATE APGM &PLX. &SYS. &RES. &MNR.
*
*TEMPLATE APGP &PLX. &RES.
*TEMPLATE APGPM &PLX. &RES. &MNR.
*
*TEMPLATE MTR &SYS..&SUB.
*TEMPLATE MTRM &MNR.
*
*TEMPLATE SYS &PLX..&RES.
*TEMPLATE SYSM &PLX..&RES. &MNR.
*
*TEMPLATE SYG &PLX..&RES.
*TEMPLATE SYGM &PLX..&RES. &MNR.
*
*TEMPLATE GRP &RES. GRP
*TEMPLATE GRPM &RES..&MNR. GRPM
*
*TEMPLATE HEARTBEATM &PLX..&RES. &MNR. &EVT. &DATE..&TIME.
*
*TEMPLATE WTORM &MNR. &EVT.
*TEMPLATE TAPEM &MNR. &EVT.
*
*TEMPLATE CFM &PLX..&RES. &MNR. &EVT.
*TEMPLATE CDSM &RES. &MNR. &EVT.
*TEMPLATE ETRM &MNR. &EVT.
*TEMPLATE SYSPLEXM &PLX..&RES..&MNR. &EVT.*
*TEMPLATE USER &STR. &DATE. &TIME
*TEMPLATE USERM &MNR. &PLX. &SUB. &DATE. &TIME. &EVT.
*********************************************************************
```

# Appendix C. Miscellaneous Information

This section tells you how to do the additional installation tasks involved in using the enterprise monitoring functions of SA z/OS.

## Running Two NetViews on the NMC Focal Point System

If your focal point system runs one NetView for automation (Automation NetView) and another NetView for networking (Networking NetView) that includes an NMC focal point system, you must install SA z/OS on both NetViews. The SA z/OS installation on the NetView used for networking involves only a subset of SA z/OS code, called an SA z/OS satellite, and fewer installation steps are required.

Where the Networking NetView is an enterprise monitoring focal point, the SA z/OS NetView's DSI6INIT Parm should specify the Networking NetView on the same system as its focal point. The focal point needs to receive heartbeats from the SA z/OS domain on the same system to set the necessary RODM focal point fields.

Installation of an SA z/OS satellite is covered as an optional step. See "Step 25: Install an SA z/OS Satellite" on page 134.

## Users and RODM Authorization

When RODM is installed on your system, it is necessary to authorize users and applications to access RODM services. This authorization is accomplished using RACF or an equivalent security application. See *Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide* for details about specifying RODM authorization. This section describes any additional user IDs that must be created for system operations enterprise monitoring and indicates whether they require RODM authorization.

*Table 25. RODM Authorization for user IDs*

| User ID | RODM Authorization Required? |
|---|---|
| NetView Graphic Monitor Facility operators | No |
| SA z/OS operators | Yes |
| User ID for bulk updates from NetView (specified in AOFRODM) | Yes |
| User ID for GMFHS to connect to RODM (defined when you install GMFHS and RODM) | Yes |

Graphic Monitor Facility Host Subsystem (GMFHS) operator IDs are usually created to be the same as NetView operator IDs so that operators can use the same ID and password to log on to GMFHS as they use to log on to NetView. RODM authorization is not required for use of GMFHS, but the IDs may require authorization for other purposes such as using RODMVIEW.

**Note:** If you assign an GMFHS operator ID of `OPER1` on the NMC focal point system, GMFHS automatically uses the same GMFHS operator ID on other NetViews in the enterprise as the target for commands.

In addition to logging on to GMFHS, operators using system operations enterprise monitoring need to log on to SA z/OS. You may choose to use the same set of IDs for SA z/OS as you do for NetView and GMFHS. However, SA z/OS IDs must be authorized to RODM. Because an ID can only be used to connect to RODM from one application at a time, you should create a unique system operations ID for each operator who connects to RODM from another application.

## Verifying Installation of SA z/OS Satellite (Optional)

You should now test your Networking NetView (with added system operations satellite). An outline procedure for this is:

1. Schedule a testing period. You will require your focal point system and expertise on how the Networking NetView should behave.
2. Shut down your Networking NetView. This means you no longer have any network automation.
3. Start your Networking NetView with the SA z/OS satellite.
4. Check that it initializes without error.
5. Check that your Networking NetView still works.
6. Start the NetView with the satellite installed and the SA z/OS topology manager configured. At this point, the SA z/OS topology manager should automatically contact all defined target sysplexes, retrieve their configuration information and create corresponding objects in RODM. Finally it will run the BLDVIEWS statements that you have defined for each sysplex. These will create views in RODM allowing you to see the objects created by the SA z/OS topology manager.
7. Start an NMC server connected to the focal point system and then connect to it from an NMC client. You should see the views defined by your BLDVIEW statements. These should contain objects representing the automated resources on the target sysplexes. There should be a green heartbeat icon for each active target sysplex.
8. If you select an icon representing an automated resource and right-click, you should see SA z/OS commands on its context menu. Select INGINFO and see that the command is issued properly.
9. Shut down the new Networking NetView, bring up the former one, and plan for production cutover.

# Enabling SA z/OS Support for Extended Multiple Console Support (EMCS)

This section describes how to set up extended multiple console support and also describes EMCS's restrictions and limitations.

**Note:** EMCS support is mandatory for the successful operation of SA z/OS.

## Setting Up EMCS

- Assign each CSSIR task a unique TSKID name in the sysplex. To do this, specify a unique name for SSIname in the the NetView style sheet that resides in the DSIPARM data set.
- Update the synonym %AOFSIRTASK%, in member AOFMSGSY, to reflect the new name of the NetView CNMCSSIR task.
- Code MSGIFAC=SYSTEM on both the NetView task (in DSIDMNK) and the SSI task (in the procedure itself).
- Add the AOCGETCN command to the initial CLIST of your operator profiles.
- Switch on the SA z/OS global variable AOF_EMCS_AUTOTASK_ASSIGNMENT, to assign an autotask to EMCS consoles.

## EMCS Restrictions and Limitations

- It is highly recommended that you code MSGIFAC=SYSTEM on both the NetView and SSI tasks. However if you need to code MSGIFAC=SSIEXT you will be unable to manage the SSI using SA z/OS.
- There must be only one NetView running SA z/OS in each machine.
- Do not:
  - Use route codes to route messages to any NetView task console
  - Deactivate the action message retention facility (AMRF) (by coding COM=’K M,AMRF=N’ in the COMMNDxx member of SYS1.PARMLIB)
  - Change the MSCOPE setting on the xxxCSSIR task/console
  - Define the AUTO attribute for any NetView task/console under the RACF OPERPARMS
  - Define a SAF OPERPARM definition for extended MCS console authority to anything other than MASTER

Violation of these restrictions will cause unpredictable results.

# Appendix D. Processor Operations Sample

This section provides a sample that demonstrates how to use processor operations for a NetView connection.

## Host VTAM Definitions for a NetView Connection through an OSA Adapter

Channel attached Major node:

```
*=====================================================================*
*    VTAM XCA MAJNODE OVER OSA                                        *
*=====================================================================*
IPSLXCA1 VBUILD TYPE=XCA
IPSLPCA1 PORT  ADAPNO=1,                                               +
               CUADDR=110E,                                            +
               MEDIUM=RING,                                            +
               SAPADDR=04,                                             +
               TIMER=60
*** ___________________________________________________________________
***
IPSLXTG1 GROUP DIAL=YES,                                               +
               DYNPU=YES,                                              +
               DYNPUPFX=PX,                                            +
               ANSWER=ON,                                              +
               AUTOGEN=(16,L,P),                                       +
               CALL=INOUT,                                             +
               ISTATUS=ACTIVE
```

Switched Major Node for a processor operations support element (SE)

```
*=====================================================================*
*    VTAM SW MAJNODE                                                  *
*=====================================================================*
***
IPSLSWN2 VBUILD TYPE=SWNET,MAXNO=1,MAXGRP=1
*** ___________________________________________________________________
***

IPSL1T00 PU    ADDR=C1,                                                +
               IDBLK=05D,                                              +
               IDNUM=E0000,                                            +
               CPNAME=IPSL1T00,                                        +
               MAXOUT=7,                                               +
               MAXPATH=1,                                              +
               MAXDATA=265,                                            +
               PUTYPE=2,                                               +
               DISCNT=NO,                                              +
               VPACING=0,                                              +
               PACING=0,                                               +
               IRETRY=YES,                                             +
               PASSLIM=1,                                              +
               ISTATUS=ACTIVE,                                         +
               MODETAB=AMODETAB,                                       +
               DLOGMOD=D4A32782,                                       +
               USSTAB=USSCP,                                           +
               SSCPFM=USSSCS
IPSLT0   PATH  DIALNO=010400203529D9CC,GRPNM=IPSLXTG1,CALL=INOUT
*** ___________________________________________________________________
***
IPSLT000 LU    LOCADDR=0,DLOGMOD=#INTER                ProcOps localLU
```

```
*** ___________________________________________________________________
***
IPSL1T01 PU    ADDR=C1,                                                +
               IDBLK=05D,                                              +
               IDNUM=D0000,                                            +
               CPNAME=IPSL1T01,                                        +
               MAXOUT=7,                                               +
               MAXPATH=1,                                              +
               MAXDATA=265,                                            +
               PUTYPE=2,                                               +
               DISCNT=NO,                                              +
               VPACING=0,                                              +
               PACING=0,                                               +
               IRETRY=YES,                                             +
               PASSLIM=1,                                              +
               ISTATUS=ACTIVE,                                         +
               MODETAB=SNAMODET,                                       +
               DLOGMOD=D4A3278A,                                       +
               USSTAB=USSCP,                                           +
               SSCPFM=USSSCS
IPSLT1   PATH  DIALNO=010400203529D9CC,GRPNM=IPSLXTG1,CALL=INOUT
*** ___________________________________________________________________
***
IPSMT100 LU    LOCADDR=0,MODETAB=MTLU6,DLOGMOD=#INTER   Support Element
IPSMT101 LU    LOCADDR=2                                3270 SESSION
IPSMT102 LU    LOCADDR=3                                3270 SESSION
```

# Appendix E. Migration Information

This appendix provides information about migrating to SA z/OS 3.2 from earlier releases. The actions that are required depend on which release you are migrating from, as follows:

- From SA z/OS 3.1 you need to perform the actions in "Migrating to SA z/OS 3.2 from SA z/OS 3.1"
- From SA z/OS 2.3 you need to perform the actions in both of the following:
  1. "Additional Actions when Migrating from SA z/OS 2.3" on page 212
  2. "Migrating to SA z/OS 3.2 from SA z/OS 3.1"

## Migrating to SA z/OS 3.2 from SA z/OS 3.1

This section describes actions you must take to migrate to SA z/OS 3.2 from SA z/OS 3.1. If you are migrating from SA z/OS 2.3, begin with "Additional Actions when Migrating from SA z/OS 2.3" on page 212.

### Migration Steps from SA z/OS 3.1

Complete the following steps to migrate to SA z/OS 3.2:

Step 1. Install the APARs OA18432 (SA z/OS 2.3) and OA19059 (SA z/OS 3.1) and open the customization dialog before converting to a SA z/OS 3.2 policy database in step 3.

Step 2. Install the compatibility APARs OA20402 (SA z/OS 2.3) and OA20403 (SA z/OS 3.1) before migrating to SA z/OS 3.2 using a SA z/OS 3.2-built configuration on a system running SA z/OS 2.3 or SA z/OS 3.1. This allows for a mixed environment.

Step 3. Make a copy of your V2.*n* or V3.1 policy database and edit it with the SA z/OS 3.2 customization dialog. This converts it to a V3.2 policy database. For more information, see "Chapter 12. Conversion Function" in *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

Step 4. Read through "Migration Notes and Advice" before migrating to SA z/OS 3.2.

Step 5. Build the configuration files from the policy database. For more information, see "Chapter 8. Building and Distributing Configuration Files" in *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

Step 6. Load the build files on the designated system. For more information, see "Step 17B: Distribute System Operations Control Files" on page 124 and "Chapter 8. Building and Distributing Configuration Files" in *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

### Migration Notes and Advice

This section contains details of various aspects of migration that you should be aware of. Make sure that you read through this section before migrating to SA z/OS 3.2. It contains the following:

- "Post-SMP/E Steps" on page 202
- "Startup and Shutdown Command Definitions" on page 202
- "NetView Automation Table Migration" on page 202
- "Resource Notification" on page 203

- “Improved WTOR Processing” on page 203
- “Improved Automation Flag Processing” on page 205
- “Changes to CICS Short-on-Storage Recovery” on page 211
- “Move Group Behavior” on page 206
- “Changes to Defaults Set via the NetView Style Sheet” on page 206
- “Changes in Message Delivery to SA z/OS” on page 207
- “CICS Monitoring” on page 207
- “Enhanced Reporting” on page 207
- “NMC Migration” on page 207
- “I/O Operations Migration” on page 210

See also Appendix F, “IMS Automation Migration,” on page 221

## Post-SMP/E Steps

You must review the following standard installation steps and, if necessary, carry them out:

1. “Step 4A: Update PROG*xx*” on page 81
2. “Step 4B: Update SCHED*xx*” on page 82
3. “Step 4D: Update LPALST*xx*” on page 82
4. “Step 4E: Update LNKLST*xx*” on page 83
5. “Step 6: Customize SYS1.PROCLIB Members” on page 87
6. “Step 7F: Add the INGRXFPG REXX Function Package” on page 96
7. “Step 10A: Customizing HSAPRM*xx*” on page 109
8. “Step 13A: Allocate Libraries for the Dialogs” on page 113
9. “Step 16: Compile SA z/OS REXX Procedures” on page 122 (if necessary)
10. “Step 22: Check for Required IPL” on page 131
11. “Step 33: Customizing I/O Operations” on page 155 (if you need to specify communication via IP)

## Startup and Shutdown Command Definitions

&EHKVAR1 variables in defined startup or shutdown commands are no longer substituted with the value provided in the Appl Parms input field of the INGREQ command. Replace &EHKVAR1 variables in startup and shutdown commands with the &APPLPARMS variable.

## NetView Automation Table Migration

If you have loaded NetView automation tables (ATs) in parallel, you must migrate these to the new release of SA z/OS.

If you do not exploit the dynamic automation-table build process and you want to maintain SA z/OS automation tables manually, you can create INGMSG02 based on:

1. Your current policy.

   Run the build process with the AT scope set to ENTERPRISE using your current policy in order to generate one automation table. This automation table is valid for all systems that are defined in the policy. The automation table is generated in the build output data set as member ACFEZ999. You must copy this member to INGMSG02, which will serve as the basis for further manual changes.
2. An SA z/OS-provided automation table member named INGMBASE.

This automation table was generated based on the sample policies and shipped with SA z/OS 3.2 when it became generally available. INGMBASE therefore contains certain AT entries that are specific to the sample add-on policies. For example, INGMBASE contains sample policy-specific job names or job-specific automation table entries. You must copy INGMBASE to INGMSG02, which will serve as the basis for further manual changes. Adapt INGMSG02 to your environment.

Although INGMBASE will not be serviced, SA z/OS will provide member INGATCHG, which contains the documentation of automation table-related service changes. Use INGATCHG to maintain your version of INGMSG02.

INGATCHG lists all APARs that affect the build of automation tables. For each APAR it documents the APAR number, the APAR description, the affected message and the change of the automation table entry as shown in Figure 27.

```
========================================================================
APAR Number: OA13642                                               @01A
------------------------------------------------------------------------
Problem: Automation does not restart IMS after it has been FORCED
         indicated by message
         'IEF743I jobname FORCED - CODE SA22 - IN ADDRESS SPACE asid'

Affected message: IEF743I

Replaced Automation Table entry

IF MSGID = 'IEF743I' & TOKEN(2) = SVJOB THEN
EXEC(CMD('AOCFILT ' SVJOB ' TERMMSG FINAL=YES,JOBNAME=' SVJOB)
ROUTE(ONE %AOFOPGSSOPER%));

by

IF MSGID = 'IEF743I' & TOKEN(2) = SVJOB THEN
EXEC(CMD('AOCFILT ' SVJOB ' TERMMSG JOBNAME=' SVJOB ',FINAL=YES,
CODE1=' SVJOB ',CODE2=SA22') ROUTE(ONE %AOFOPGSSOPER%));
```

*Figure 27. Sample INGATCHG Entry*

**Note:** The advanced automation option AOFSMARTMAT must be set to 0 or 1 so that INGMSG02 is loaded.

## Resource Notification

You must add the values NMC and SDF to the new **Inform List** field in the AUTOMATION OPTIONS policy of the SDF entry type and the RESOURCE INFO policy of the new sysplex defaults XDF entry type to ensure that NMC and SDF are notified as before, if they are used. Such system defaults must be linked to every system, and sysplex defaults must be linked to every SA z/OS subplex. Otherwise, NMC and SDF will no longer be informed and updated.

## Migration of Automation Operators

**Prerequisites for SA z/OS Initialization:** Make sure that the automated operator function EVTOPER is active, otherwise SA z/OS will not initialize. To achieve this, define AUTEVT1 as the primary and AUTEVT2 as the secondary automation operator for this operator function via the Automation Operators policy object.

## Improved WTOR Processing

Improved processing of WTORs has affected the following:

- "Assign WTOR Delete Messages to SYSOPER" on page 204
- "Shutdown Command to NetView" on page 204

- “Type Specifications for WTORs”
- “Reply Definitions in the MESSAGES/USER DATA Policy Item”
- “Resynchronization of WTOR Data after Communication Task Abend”
- “Executing Task of OUTREP” on page 205
- “Automation Status File” on page 205
- “Routines” on page 205

**Assign WTOR Delete Messages to SYSOPER:** The WTOR delete messages IEE400I and IEE600I should be assigned to the SYSOPER.

To assign the IEE400I and IEE600I messages to SYSOPER, use the appropriate Automation Operators policy object in the customization dialog. In the OPERATORS policy item for SYSOPER specify the message IDs IEE400I and IEE600I as messages for this operator.

**Shutdown Command to NetView:** In earlier SA z/OS releases, if a shutdown command to a NetView subsystem was defined in the following format it was issued as a reply to a NetView subsystem with the specified domain ID:

```
CLOSE option,domain
```

To prevent the misinterpretation of a second option as the domain ID, this function has been removed in SA z/OS 3.2. You can now directly define the shutdown command to NetView as a reply in the policy database.

**Type Specifications for WTORs:** Earlier releases of SA z/OS also replied to WTORs with a priority of SECONDARY after replying to all WTORs with a priority of PRIMARY. SA z/OS 3.2 does not reply to WTORs that have been defined with a priority of SECONDARY. You must therefore check the code definitions for type WTORS of all entries for the correct status definitions, that is, change the priority to PRIMARY for those WTORs that should be replied to.

**Reply Definitions in the MESSAGES/USER DATA Policy Item:** Code definitions for a message ID of an entry in the MESSAGES/USER DATA policy item can be used by generic routines ISSUEREP, ISSUECMD or ISSUEACT to check for matching codes.

In previous releases of SA z/OS, the value returned for a code match was used by ISSUEREP and ISSUECMD differently:

- ISSUEREP used the returned value as the reply to be issued for a received WTOR.
- ISSUECMD used the returned value to select a defined command.

In SA z/OS 3.2 the value returned is only assumed as the reply to be issued for a received WTOR if there are no additional REPLY or CMD definitions. Otherwise the value returned is used to select defined commands or replies.

Make sure that code definitions that have replies as the value returned are not combined with CMD or REPLY definitions for the same message ID of an entry.

**Resynchronization of WTOR Data after Communication Task Abend:** The following statement has been removed from the automation table:

```
IF MSGID = 'IEE824E' THEN
EXEC(CMD('AFTER 00:00:30 AOFRSGOR IEE824E') ROUTE(ONE %AOFOPWTORS%))
```

If the stored WTOR data should be resynchronized because of a communication task abend, an appropriate RESYNC command has to be defined in the automation policy for the IEE824E message.

**Executing Task of OUTREP:** OUTREP is no longer executed by one of the tasks that are defined as the list for synonym %AOFOPWTORS%, but by one of the tasks that are defined as the list for synonym %AOFOPSYSOPER%.

Calls to OUTREP in overwrite definitions for automatically-generated automation table statements, or to OUTREP in user-defined automation tables have to be changed accordingly.

**Automation Status File:** SA z/OS 3.2 no longer uses the automation status file to store and provide reply identifiers from outstanding WTORs of subsystems.

SA z/OS 3.2 provides the AOFRASFR command to delete all obsolete reply identifiers from the automation status file. AOFRASFR has to be called without any parameters. After it successfully finishes, it issues the completion message AOF099I FUNCTION COMPLETED.

**Routines:** The AOFRSROR routine is no longer part of SA z/OS. Use the SUBSWTOR task global variable instead to retrieve the reply ID of the outstanding WTOR of an application. This variable is available after the automation flag for this application has been checked by AOCQRY.

## Improved Automation Flag Processing

Improved processing of automation flags has affected the following:

- "DISPLAY Assist Mode"
- "Automation Flags"
- "Evaluation of Disable Times for Automation Flags"
- "Automation Flag Exits" on page 206

**DISPLAY Assist Mode:** SA z/OS 3.2 no longer supports the DISPLAY assist mode. DISPLAY assist mode settings in the automation policy are automatically converted to the automation flag value LOG.

Although the DISPLAY assist mode settings in the automation policy are also no longer provided on systems running an earlier version of SA z/OS in a mixed environment where the automation policy has been built by SA z/OS 3.2, the assist mode on these systems can still be set to DISPLAY during run time with the SETASST command.

On systems running SA z/OS 3.2, use the new INGMDFY command to modify start or stop actions for the next startup or shutdown of a subsystem.

**Automation Flags:** The automation flag evaluation process has been changed to make it more comprehensible.

For details see the description of the AOCQRY common routine in *IBM Tivoli System Automation for z/OS Programmer's Reference*.

**Evaluation of Disable Times for Automation Flags:** The automation policy allows you to define times when automation should be disabled.

Although an automation flag may be set to YES, it is possible to turn off automation at certain times using the Automation Flag policy item in the customization dialog.

When specifying the day, beginning and end time for a disable time, you can specify an end time that is earlier than the begin time, for example, a begin time of 7 p.m. and a end time of 10 a.m..

In SA z/OS 3.1, the evaluation of time interval specifications with an end time earlier than the begin time was inconsistent. Evaluation of these time intervals has been improved in SA z/OS 3.2 to make it more predictable and plausible.

*Changes in SA z/OS 3.2:* In SA z/OS 3.2, both the weekday and time specification are checked when evaluating disable times for automation flags. Time interval specifications with an end time that is earlier than the begin time are evaluated for different day specifications. This is demonstrated in the following examples for the same Begin time of 20:00 and End time of 06:00:

| Day | Begin | End | Resulting Time Period | Changed Behvavior? |
|---|---|---|---|---|
| * | 20:00 | 06:00 | Every day from 08:00 until the next morning at 06:00 | No |
| MO | 20:00 | 06:00 | On a Monday from midnight until 06:00, and from 20:00 until midnight | Yes |
| MF | 20:00 | 06:00 | Beginning on Monday at midnight until 06:00, then each day from 20:00 until the next morning at 06:00, up to midnight on Friday | Yes |
| SS | 20:00 | 06:00 | Beginning on Saturday at midnight until 06:00, then from 20:00 until Sunday 06:00, and then from 20:00 until midnight on Sunday | Yes |

The DISPSCHD command can be used to list disable automation time intervals during runtime.

**Automation Flag Exits:** The invocation process of automation flag exits is based on the automation flag evaluation process, which has been changed in SA z/OS 3.2.

The resource parameter of automation flag exits now contains the fully-qualified resource name of the resource that the automation flag value has been requested for. It no longer contains the name of the resource, that the automation flag value is checked for during the flag evaluation process.

For details see the description of Flag Exits in *IBM Tivoli System Automation for z/OS Customizing and Programming*.

## Move Group Behavior

Move groups have been extended to provide optional serial behavior. This means that if you select this behavior, you can remove any scaffolding you had in place (MakeAvailable/WhenObservedDown relationships or proxy resources, or both) that you needed to make earlier releases provide this behavior.

## Changes to Defaults Set via the NetView Style Sheet

WTOs and WTORs issued by SA z/OS.

The defaults for the following advanced automation option (AAO) variables have changed:

- AOF_INIT_MCSFLAG = 00000000 (from 00001000)
- AOF_INIT_SYSCONID = (from 01)
- AOF_INIT_ROUTCDE = 01000000 (from 10000000)

AOFSTYLE now contains the following overrides for NetView settings:

- MVSPARM.MSGIFAC = SYSTEM
- INIT.TIMER = No

## Changes in Message Delivery to SA z/OS

SA z/OS 3.2 no longer issues the MN JOBNAMES command in order to the required IEF403I, IEF404I and IEF450I messages.

During initialization the current setting for MONITOR is evaluated using the command:

```
DISPLAY OPDATA,MONITOR
```

If JOBNAMES is found to be OFF SA z/OS issues the following command to turn it on:

```
SETCON MONITOR,JOBNAMES=(ON,NOLOG)
```

The messages that are produced by JOBNAMES monitoring are not logged. If you want any other setting you have to add an appropriate SETCON command to the COMMND*xx* PARMLIB member.

## CICS Monitoring

The CICS link and health monitoring in SA z/OS 3.1 is supported in SA z/OS 3.2 but will not be supported in future releases of SA z/OS. You should therefore migrate to the new event-based CICSPlex monitoring that has been introduced in SA z/OS 3.2.

For more details see “How to Monitor Applications” in *IBM Tivoli System Automation for z/OS Customizing and Programming*.

## Enhanced Reporting

To write SA z/OS SMF records to the SMF log, ensure that the SMFPRM*xx* member in SYS1.PARMLIB is set up to collect type 114 SMF records, see “Step 4H: Update SMFPRM*xx*” on page 84.

For SA z/OS to write to the SMF log, specify SMF in the Inform List of the APLs, APGs and MTRs that you want SMF records written for.

For more details see also “Availability and Recovery Time Reporting” in *IBM Tivoli System Automation for z/OS Customizing and Programming*.

## NMC Migration

The following changes have been made to the policy database (note that the add-on sample *NMC has been updated):

- Heartbeat slave operator HBSLV (AUTHBSLV) should be added to the NMC-related AUTO OPERATORS in the policy database. This operator is used for heartbeat slave processing.

- INGPOST slave operator POSTSLV (AUTPOSTS) should be added to the NMC-related AUTO OPERATORS in the policy database. This operator prevents deadlock situations on the NMC focal point when the command handler is used.

**NMC Secondary Focal Point Domains:** It is no longer possible to define secondary NMC backup focal points in the customization dialog.

When running with SENDALERT=YES all forwarding information is defined within NetView.

When running with SENDALERT=NO primary NMC focal points are supported.

**NMC Exploitation:** The following migration advice is for installations that use SA z/OS NMC exploitation. Skip this section if you do not use NMC.

*In General:* **Inform List** can be used to exclude resources from being displayed on the NMC client after *all* systems (NMC focal point, NMC target sysplex) are running SA z/OS 3.2.

A mixture of systems running SA z/OS 3.2 and earlier versions or releases of SA z/OS requires the **Inform List** field to be set to ’NMC’ for *all* resources.

More details can be found below and in the ’readme.txt’ file in the SA z/OS NMC workstation code (filenames: INGNMCZP or INGNMCTZ).

**Migration Scenarios:**

*Scenario 1:* Consider the following scenario:

- The NMC focal point is running SA z/OS 3.1
- The first NMC target system is running SA z/OS 3.2
- The first NMC target system has an SA z/OS 3.2 automation manager
- The other NMC target systems are running SA z/OS 3.1 or earlier

Where *first* means the first system that is mentioned in SYSPLEX statement in INGTOPOF, and *other* means any systems that are mentioned after the first one in the SYSPLEX statement in INGTOPOF.

This scenario requires you to specify `NMC` in the SA z/OS 3.2 customization dialog for *all* resources of type APL, APG, and MTR in their **Inform List** field. You can use the System Defaults and Sysplex Defaults entry types to do this.

**Remark:** An NMC target system running SA z/OS 3.2 cannot cooperate with an SA z/OS 3.1 automation manager (because there is no **Inform List** field in SA z/OS 3.1).

*Scenario 2:* Consider the following scenario:

- The NMC focal point is running SA z/OS 2.3 or earlier
- The NMC target system is running SA z/OS 3.2

This scenario is *not* supported.

The NMC SA z/OS 3.2 target system needs an NMC focal point running SA z/OS 3.1 or SA z/OS 3.2.

*Scenario 3:* Consider the following scenario:

- The NMC focal point is running SA z/OS 3.2

- The first NMC target system is running SA z/OS 3.1
- The first NMC target system has an SA z/OS 3.1 automation manager
- The other NMC target systems are running SA z/OS 3.1 or earlier

Where *first* means the first system that is mentioned in SYSPLEX statement in INGTOPOF, and *other* means any systems that are mentioned after the first one in the SYSPLEX statement in INGTOPOF.

This scenario is supported.

*Scenario 4:* Consider the following scenario:

- The NMC focal point is running SA z/OS 3.2
- The first NMC target system is running SA z/OS 3.2
- The first NMC target system has an SA z/OS 3.2 automation manager
- The other NMC target systems are running SA z/OS 3.1 or earlier

Where *first* means the first system that is mentioned in SYSPLEX statement in INGTOPOF, and *other* means any systems that are mentioned after the first one in the SYSPLEX statement in INGTOPOF.

This scenario requires you to specify `NMC` in the SA z/OS 3.2 customization dialog for *all* resources of type APL, APG and MTR in their **Inform List** field. You can use the System Defaults and Sysplex Defaults entry types to do this.

*Scenario 5:* Consider the following scenario:

- The NMC focal point is running SA z/OS 3.2
- The first NMC target system is running SA z/OS 3.1
- The first NMC target system has an SA z/OS 3.1 automation manager
- The other NMC target systems are running SA z/OS 3.2 or earlier

Where *first* means the first system that is mentioned in SYSPLEX statement in INGTOPOF, and *other* means any systems that are mentioned after the first one in the SYSPLEX statement in INGTOPOF.

This scenario is *not* supported.

**Remark:** An NMC target system running SA z/OS 3.2 cannot cooperate with an SA z/OS 3.1 automation manager (because there is no **Inform List** field in SA z/OS 3.1).

*Scenario 6:* Consider the following scenario:

- The NMC focal point is running SA z/OS 3.2
- The first NMC target system is running SA z/OS 3.1
- An NMC target system (first or other) with SA z/OS 3.2 automation manager
- The other NMC target systems are running SA z/OS 3.2 or earlier

Where *first* means the first system that is mentioned in SYSPLEX statement in INGTOPOF, and *other* means any systems that are mentioned after the first one in the SYSPLEX statement in INGTOPOF.

This scenario requires you to specify `NMC` in the SA z/OS 3.2 customization dialog for *all* resources of type APL, APG and MTR in their **Inform List** field. You can use the System Defaults and Sysplex Defaults entry types to do this.

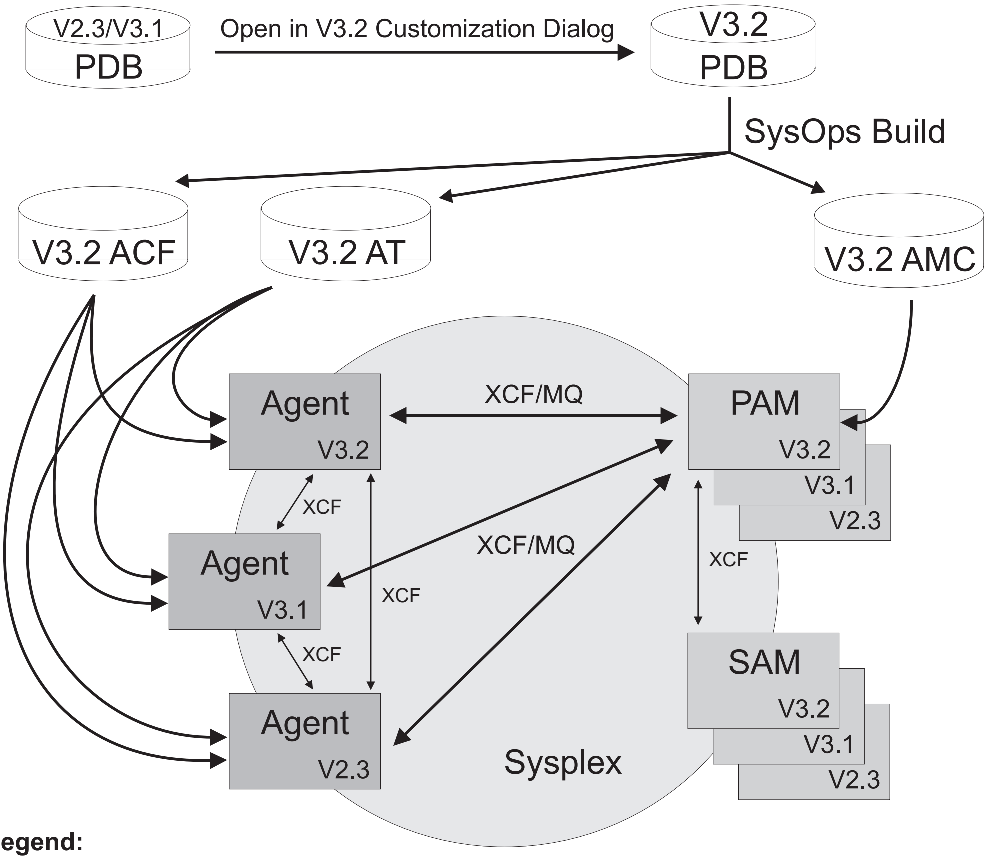
### I/O Operations Migration

With SA z/OS 3.2 I/O Operations requires CSA user key storage. If you are running z/OS V1.8 or higher, ensure that the parameter VSM ALLOWUSERKEYCSA is set to YES in the DIAG*xx* parmlib member. See also "Step 33: Customizing I/O Operations" on page 155.

## Coexistence of SA z/OS 3.2 with Previous Releases

It is not expected that you will cut over all your systems at the same time from previous releases to SA z/OS 3.2. This means that you may be running different releases at the same time.

SA z/OS 3.2 systems can coexist with SA z/OS 3.1 and SA z/OS 2.3 systems in the same sysplex. Figure 28 illustrates this: it shows a sysplex with three automated systems and a separate automation manager (and its secondary).



**Legend:**
PDB: Policy database
ACF: Automation agent's automation configuration files
AT: NetView automation table
AMC: Automation manager configuration files

*Figure 28. Coexistence of SA z/OS 3.2, SA z/OS 3.1 and SA z/OS 2.3*

Any policy database created by a earlier version of the customization dialog (that is, earlier than SA z/OS 3.2) is automatically converted into the SA z/OS 3.2 format when the policy database is opened the first time using the SA z/OS 3.2 customization dialog.

The automation configuration files (ACF) built by the SA z/OS 3.2 customization dialog can be used by any automation agent running either SA z/OS 3.2, SA z/OS 3.1, or SA z/OS 2.3. In other words, the ACF fragments that are built are compatible, so they can be used by any automation agent running an earlier version of SA z/OS.

The NetView automation table (AT) created by the SA z/OS 3.2 customization dialog can be used by automation agents running either SA z/OS 3.1 or SA z/OS 2.3, but an additional automation table is required for compatibility with SA z/OS 3.2. This compatibility automation table includes all the statements used in SA z/OS 3.1 or SA z/OS 2.3 that are no longer built by SA z/OS 3.2.

This compatibility automation table is provided in member INGMSG32 and must be declared with the SA z/OS 3.2 customization dialog in the SYSTEM INFO policy item of the system policy object. It must be specified in the Automation Table(s) field, ahead of INGMSG01. You can customize member INGMSG32 according to your needs.

In a sysplex (that is, the same XCF group) automation agents running SA z/OS 3.2, SA z/OS 3.1 or SA z/OS 2.3 can communicate with an SA z/OS 3.2 automation manager. The communication is either via XCF or WebSphere MQ. The automation agents communicate with each other via XCF.

## Nested Class Support

The function will only be available from SA z/OS 3.2 or higher. During a transition phase where SA z/OS 3.2 coexists with systems running earlier releases, you need to be very careful with defining multiple class levels. This may be appropriate, for example, for the definition of **Inform List** because this field is supported only for SA z/OS 3.2 and higher, but in other cases it should be carefully considered whether multiple classes could result in data loss on systems running earlier releases. A PTF will be available that issues a warning during load on a down level system if a class is detected that is linked to further classes.

With SA z/OS 3.2 the application type will be checked during class-instance link processing. It will only be allowed to link all classes or instances to a class with no type specified, and to a class of the same type. If there are other links in a Policy Database (for example, an instance of type CICS linked to a class of type IMS) they will still exist when migrating to this release, but when the corresponding link panel is invoked such links will be marked as invalid and need to be changed.

## Behavior Change

Consider you are operating a sysplex where some systems are running SA z/OS 3.2 and some are not. All systems are using a SA z/OS 3.2 automation policy.

If STARTAFTERIPL=NOSTART is specified, a stop request with a priority of FORCE is injected. The compound status of the sysplex application group is then determined from the compound status of its members that are running SA z/OS 3.2.

**Effects:** The most noticeable effects will be:

- SA z/OS 3.2 systems will implement STARTAFTERIPL=NOSTART via the new request-based mechanism.
- Systems will implement STARTAFTERIPL=NOSTART via the old state mechanism.

## Changes to CICS Short-on-Storage Recovery

If you are using an automatically built NetView automation table, you do not have to take any action and can skip this section.

If you are *not* using dynamically built NetView automation tables, consider the following changes:

- If you have CICS Automation active and are not using the automatically built NetView automation table, you must change your AT. Locate the AT entry for the messages DFHSM0131, DFHSM0132, DFHSM0133, and DFHSM0134:

  ```
  IF (GROUP:INGCICS)
  MSGID = 'DFHSM013n'
   THEN
   EXEC(CMD('AOCFILT * EVEEY00S ')ROUTE(ONE %AOFOPGSSOPER%));
  ```

  Change the AT entry to:

  ```
  IF (GROUP:INGCICS)
  MSGID = 'DFHSM013n'
   THEN
   EXEC(CMD('AOCFILT * EVEES100 ')ROUTE(ONE %AOFOPGSSOPER%));
  ```

  This change is only required if you want to use an ACF that is built by SA z/OS 3.2 on a lower release.

## Additional Actions when Migrating from SA z/OS 2.3

This section describes the additional actions you must take to migrate from SA z/OS 2.3 before you can proceed with the actions in “Migrating to SA z/OS 3.2 from SA z/OS 3.1” on page 201.

**Note:** You do not need to perform these actions if you are migrating from SA z/OS 3.1. For information about how to migrate from SA z/OS 3.1 see “Migrating to SA z/OS 3.2 from SA z/OS 3.1” on page 201.

The following steps are required to migrate from SA z/OS 2.3:

- “Policy Database Migration”
- “ACF Migration” on page 213
- “Automation Status File” on page 213
- “Automation Table Migration” on page 214
- “DB2 Automation: Critical Event Monitoring” on page 214
- “CICS Automation: EVESTIEX” on page 217
- “IMS Automation: Define Restart Commands in Response to Message DFS810A” on page 218
- “IMS Automation: EVISPINM” on page 219
- “Equivalents to Retired Commands” on page 219
- “Coupling Facility Structures” on page 219
- “Incompatibilities” on page 219

### Policy Database Migration

Entry type ICL, which has not been actively supported since V2.1, is no longer available.

The information that was stored in entry type DEN needs to be manually migrated. It now needs to be stored in the DB2 CONTROL policy item of APL policy objects that have an application type of DB2.

The SAF ENVIRON policy item in entry type NTW is no longer supported. The information that was stored in there now needs to be stored in entry type UET. To migrate it, you can exploit the **Migrate from ACF** option on the Data Management Menu panel of the customization dialog (see *IBM Tivoli System Automation for z/OS Defining Automation Policy* for details). This will now automatically place the SAF ENVIRON information in entry type UET.

**Note:**
Any information in entry types ICL or DEN or in policy item SAF ENVIRON of entry type NTW will be discarded. So if required by your environment, make sure you migrate it as described in “Policy Database Migration” on page 212.

## ACF Migration

SA z/OS 3.2 requires the automation control file and the associated automation manager configuration file to be consistent and to be built from the policy database in one step.

Step 1. If you have OMEGAMON installed and you want to integrate monitoring information and exceptions with SA z/OS 3.2, define AOFSES*xx* automated functions mapped to AUTSES*xx* automation operators in entry type AOP. The AUTSES*xx* automation operators must be defined in the NetView DSIOPF PARMLIB member.

Step 2. For automated functions named MVSCONS*i* in entry type AOP, the MVS console ID specification is no longer available. Supply an MVS console name instead.

Step 3. Build the system operations configuration files (automation control file, automation manager configuration file, NetView Automation Tables, and MPFLST member) from the customization dialog. For more information, see *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

## Automation Status File

The layout of the automation status file has changed. How you proceed depends on whether you need to keep the information that in your V2.3 automation status file:

- If you do *not* need to keep any information that is currently in your V2.3 automation status file, simply delete it and allocate a fresh automation status file with the V3.2 layout by running the INGALLC2 sample job as described in “Step 2C: Data Sets for Automation Agents” on page 78.
- If you do need to keep the information that is currently in your V2.3 automation status file, proceed as follows:
  1. Make a backup copy of your V2.3 automation status file.
  2. Delete the original data set.
  3. Reallocate a data set for the automation status file by running the INGALLC2 sample job as described in “Step 2C: Data Sets for Automation Agents” on page 78. This will use the new layout.
  4. Use the AOFEXASF migration utility that is shipped with SA z/OS 3.2 to copy the status information from the backup copy of your V2.3 automation status file to the newly allocated V3.2 automation status file.

## Automation Table Migration

The predefined automation table fragment INGMSG02 is no longer shipped. If you do not exploit the dynamic automation table build process and you want to maintain SA z/OS automation tables manually, you can create INGMSG02 based on:

- **Your current policy:**

  Run the build process with the AT scope set to ENTERPRISE using your current policy in order to generate one automation table. This automation table is valid for all systems that are defined in the policy. The automation table is generated in the build output data set as member ACFEZ999. You must copy this member to INGMSG02, which will serve as the basis for the further manual changes that follow.
- **The SA z/OS-provided automation table member, INGMBASE:**

  This table is generated, based on the sample policies, and delivered with SA z/OS 3.2. INGMBASE therefore contains certain AT entries that are specific to the sample policies. For example, INGMBASE contains sample policy-specific job names or job-specific automation table entries.

  You must copy INGMBASE to INGMSG02, which will serve as the basis for the further manual changes that follow. Adapt INGMSG02 to your environment.

Although INGMBASE will not be serviced, SA z/OS will provide member INGATCHG, which contains the documentation of automation table-related service changes. Use INGATCHG to maintain your version of INGMSG02.

INGATCHG lists all APARs that affect the build of automation tables. For each APAR it documents the APAR number, the APAR description, the affected message and the change of the automation table entry.

Note that the advanced automation option AOFSMARTMAT must be set to 0 or 1 so that INGMSG02 is loaded.

## DB2 Automation: Critical Event Monitoring

Critical event monitoring in DB2 automation handles specific critical events that may occur during normal day to day running of DB2. In SA z/OS 3.1, the processing of some DB2 messages has been moved from special DB2 automation routines to generic routine ISSUECMD. This provides more flexibility in customizing the DB2 automation.

- The relating automation table statements for triggering the automation actions by incoming messages can now be overridden.
- Some of the DB2 messages can now be automated even if DB2 is not defined as an application of DB2.
- Automation table statements to the relating messages are only created when commands to be issued are defined in the automation policy.

In the following, the changes in the automation of the DB2 messages concerned are described in detail.

- **DSNP007I: Dataset extension failed**

  **DSNT500I: Resource unavailable**

  **DSNT501I: Resource unavailable**

  **Situation in V2.3:** In SA z/OS 2.3, specific DB2 routines have been triggered by each of these messages via an appropriate entry in the automation table INGMSG02. The DB2 routines used code specifications to message ID

DATABASE in policy item MESSAGES/USER DATA of the DB2 entry or MVSESA to check, if the code definitions matched the triggering message. The code definitions of the automation control file have been silently completed by database IDs DSNDB01 and DSNDB06. If a match has been found for a code specification with value returned NULL and the message has been issued by a BATCH job, the defined commands to the message ID of the triggering message has been issued. For non BATCH jobs, only an alert has been issued.

**Changes in V3.1:** In SA z/OS 3.1, ISSUECMD is called instead of the DB2 specific routines. The standard generic routine is smart enough to provide all functionality for processing these messages. For a greater granularity the code specifications are expected to be defined to the message ID of the triggering message instead of message ID DATABASE. In SA z/OS 3.1, ISSUECMD is called with code specifications as parameters. The code values are extracted from the message text. In case of message DSNP007I the data set name is passed as CODE1, the return code is passed as CODE2 and the connection ID is passed as CODE3 value. In case of message DSNT500I or DSNT501I the name is passed as CODE1, the reason is passed as CODE2 and the type is passed as CODE3. If a match is found to the ID of the triggering message in policy item MESSAGES/USER DATA, the returned value is used to select and issue the relating commands defined in the automation policy.

**Migration:** To migrate the automation processing of these messages from SA z/OS 2.3 ~~to SA z/OS 3.1~~ unchanged as far as possible, copy the code definitions with NULL as returned value from message ID DATABASE to the relating message IDs. Adapt the code definitions to the code values passed as parameters when calling ISSUECMD in the automation table by:

1. Specifying connection ID BATCH as the value for CODE3 (only for message DSNP007I)
2. Adding code definitions for database IDs DSNDB01 and DSNDB06
3. Adding an asterisk (*) as wild character at the beginning and/or end of the code values, if they do not represent the appropriate whole extracted value of the message text

Add the returned value to the code specifications as selection name to the commands defined to the relating message ID in the automation control file, if the commands should only be issued, if a code match occurs. If you still have a down level system in your installation, where you do not use the dynamic automation table build process, you have to keep the old definitions for message ID DATABASE in the automation policy, which you have copied to the message IDs of the triggering messages. Otherwise you can delete them.

- **DSNV086E: Final termination message**

  **Situation in V2.3:** In SA z/OS 2.3, automation table INGMSG02 contained several statements to the final termination message DSNV086E of DB2. Dependent on the reason code included in the message text, the status of DB2 has been changed to BROKEN or ABENDED by calling the generic routine TERMMSG with parameters FINAL=YES and BREAK=YES or ABEND=YES. In additional automation table statements labelled with INGDB2, the special DB2 routine INGRDTTH has been called in addition to generic routine TERMMSG.

  **Changes in V3.1:** In SA z/OS 3.1, the checking of the reason code in the message text has been moved from the automation table to code specifications in the automation policy. This allows you a more flexible customization of these definitions without having the need to override the relating automation table statements. In SA z/OS 3.1, the automation table statement calls the generic routine TERMMSG with parameter FINAL=YES and with code specifications, which are used to search the automation control file for an action that modifies the BREAK or ABEND parameter. Furthermore, the special DB2 routine INGRDTTH is called by TERMMSG itself, if the application having issued the triggering message is known to SA as application of subtype DB2.

**Migration:** To migrate the functionality of the automation table statements of SA z/OS 2.3 unchanged~~ to SA z/OS 3.1~~, define the following code specifications to message ID DSNV086E in policy item MESSAGES/USER DATA of entry DB2 or MVSESA in the policy database:

| Code 1 | Code 2 | Code 3 | Value Returned |
|---|---|---|---|
| *jobname* | 00F70600 | * | BROKEN |
| *jobname* | 00F70602 | * | BROKEN |
| *jobname* | 00E30105 | * | BROKEN |
| *jobname* | 00E30078 | * | BROKEN |
| *jobname* | * | * | ABENDED |

Code1 specifies the job name of DB2 which issues message DSNV086E and code2 specifies the reason code, included in the message text. Code3 is not used.

- **DSNJ002I: Switch active log data sets**

  **Situation in V2.3:** In SA z/OS 2.3, specific DB2 routine INGRD002 had been triggered by message DSNJ002I via an appropriate entry in the automation table INGMSG02. The DB2 routine tracked the occurrence of message DSNJ002I for the log dataset specified by the “Active log data set name” in the DB2 CONTROL policy item. The routine issued a command when a critical threshold was reached. The command to be issued had to be defined in the automation policy item MESSAGES/USER DATA of the DB2 resource entry to message DSNJ002I. The threshold had to be entered against the LOG minor resource for the DB2 resource.

  **Changes in V3.1:** In SA z/OS 3.1, generic routine ISSUECMD is called instead of the DB2 specific routine. ISSUECMD does not check the occurrence of the triggering message DSNJ002I. The command defined in the automation policy is issued each time, when it is triggered by message DSNJ002I for the log data set specified by the “Active log data set name” in the DB2 CONTROL policy item. Message ING115A is no longer issued.

- **DSNJ110E: Last active log data set is % full**

  **Situation in V2.3:** In SA z/OS 2.3, specific DB2 routine INGRD111 had been triggered by message DSNJ110E via an appropriate entry in the automation table INGMSG02. The DB2 routine compared the reported percentage full figure of the triggering message with the critical threshold defined in the “Log full threshold” field of the DB2 CONTROL policy item. If the value in the message exceeded this threshold, automation proceeded to issue the command defined in the automation policy item MESSAGES/USER DATA of the DB2 resource entry to message DSNJ110E with selection CRIT.

  **Changes in V3.1:** In SA z/OS 3.1, generic routine ISSUECMD is called instead of the DB2 specific routine and the command from the ACF is called without selection name.

  **Migration:** To migrate this functionality of SA z/OS 2.3 unchanged~~ to SA z/OS 3.1~~, delete the selection name of the command defined to message DSNJ110E in the policy item MESSAGES/USER DATA of the DB2 policy object.

- **DSNJ111E: All active log data sets full**

  **Situation in V2.3:** In SA z/OS 2.3, specific DB2 routine INGRD111 had been triggered by message DSNJ111E via an appropriate entry in the automation table INGMSG02. The DB2 routine issued message ING116I when the elapsed time interval because the last received message DSNJ111E has been greater than the value defined in the “Active log alerts” field of the DB2 CONTROL policy item. If furthermore the occurrence of the received DSNJ111E messages exceeded the defined threshold, automation proceeded to issue the command defined in the automation policy item MESSAGES/USER DATA of the DB2 resource entry to message DSNJ111E with selection CRIT.

**Changes in V3.1:** In SA z/OS 3.1, generic routine ISSUECMD is called instead of the DB2 specific routine and the command from the ACF is called without selection name. Message ING116I is no longer issued.

**Migration:** To migrate the functionality of SA z/OS 2.3 unchanged~~ to SA z/OS 3.1~~, delete the selection name of the command, defined to message DSNJ111E in the policy item MESSAGES/USER DATA of the DB2 policy object.

- **DSNJ115I: Archive data set could not be allocated**

  **Situation in V2.3:** In SA z/OS 2.3, specific DB2 routine INGRD115 had been triggered by message DSNJ115I via an appropriate entry in the automation table INGMSG02. The DB2 routine issued message ING117I when the elapsed time interval since the last received message DSNJ115I has been greater than the value defined in the “Log offload interval” field of the DB2 CONTROL policy item. In addition automation issued the command defined in the automation policy item MESSAGES/USER DATA of the DB2 resourc entry to message DSNJ115I with selection CRIT.

  **Changes in V3.1:** In SA z/OS 3.1, generic routine ISSUECMD is called instead of the DB2 specific routine and the command from the ACF is called without selection name. Message ING117I is no longer issued.

  **Migration:** To migrate the functionality of SA z/OS 2.3 unchanged~~ to SA z/OS 3.1~~, delete the selection name of the command, defined to message DSNJ115I in the policy item MESSAGES/USER DATA of the DB2 policy object.

- **DSN*nnnn*E: Generic alert**

  **Situation in V2.3:** In SA z/OS 2.3, specific DB2 routine INGRDREC has been triggered by messages DSN*nnnn*E via an appropriate entry in the automation table INGMSG02. The DB2 routine tracked the occurrence of incoming messages and issued either a command or a reply when a critical threshold was reached. The command to be issued had to be defined in the automation policy item MESSAGES/USER DATA of the DB2 resource entry to the relating message. The threshold had to be entered against the minor resource for the DB2 resource. The incoming messages had been captured.

  **Changes in V3.1:** In SA z/OS 3.1, generic routine ISSUECMD or ISSUEREP is called instead of the DB2 specific routine. ISSUECMD or ISSUEREP does not check the occurrence of the triggering message DSNJ002I. The command or reply defined in the automation policy is issued each time, when it is triggered by the relating message.

  **Migration:** If messages DSN*nnnn*S have to be captured even if they are not automated, define them in policy item MESSAGES/USER DATA and select “Capture” as the AT action.

## CICS Automation: EVESTIEX

CICS module EVESTIEX has been replaced by EVEPYINI. Refer to “Step 28B: Program List Table Definitions” on page 145, which shows how the DFHPLT program entry for EVEPYINI has replaced the one for EVESTIEX.

## IMS Automation: Define Restart Commands in Response to Message DFS810A

In SA z/OS 2.3, the INGMSG02 automation table included an entry for message DFS810A, belonging to label group INGIMS, and issuing the commands OUTREP and EVIEI00B. This statement caused the automation to issue a reply to an incoming DFS810A message, as defined in the automation policy MESSAGES/USER DATA of the related application, selecting the reply to the start type as the selection name. If no reply with the given start type as the selection name could be found, the start type was changed to the default value as defined in the automation policy IMS Control Region specifications. In the case of a MANUAL start type, the value provided in the Appl Parms input field of the INGREQ command was given as the response to DFS810A.

In SA z/OS 3.1 an automation table entry for message DFS810A is *only* built if a reply to this message is defined in the MESSAGES/USER DATA policy item for any of the applications in the automation policy. The automation table statement that is created issues the ISSUEREP generic routine in response to the incoming DFS810A message.

Rather than using the default start type and the start type for recovery purposes as defined in automation policy IMS Control Region specifications, the ISSUEREP routine uses NORM as the default value for the start type. Because the standard ISSUEREP routine does not handle the start type MANUAL differently to other start types, the response to DFS810A for start type MANUAL is also taken from the policy item MESSAGES/USER DATA, using MANUAL as the selection name.

A new variable, &APPLPARMS, provides the value defined as Appl Parms with the INGREQ command. It can be used when specifying commands and replies in the automation policy in response to incoming messages during the startup phase.

### Migration

To migrate from SA z/OS 2.3~~to SA z/OS 3.1~~, define the replies to DFS810A for each start type, including the default start type NORM, as the selection name in policy item MESSAGES/USER DATA.

Replies to DFS810A that are defined without a selection name are selected for each start type. Therefore replies can be defined without a selection name if they are identical for the different start types. If the MESSAGES/USER DATA policy item does not contain any definitions for message ID DFS810A, an automation table entry for this message is not included in the automation table that is automatically built. The &EHKVAR1 variable in command and reply definitions for message DFS810A must be replaced with the new &APPLPARMS variable. &APPLPARMS must also be used instead of &EHKVAR1 in IMS startup commands.

If the value specified in Appl Parms for the INGREQ command has to be issued as the response to DFS810A for a MANUAL start type, define &APPLPARMS as the command for selection name MANUAL.

Previously, a start type for recovery purposes was defined in the automation policy's IMS Control Region specifications. Now a different start type for abend recovery must be set via a status command if the status of IMS changes to ABENDING. To do this, define an appropriate INGSET SET &SUBSAPPL STARTTYPE=*starttype* command for the message ID ABENDING in the MESSAGES/USER DATA policy item for the subsystem IMS. If the MESSAGES/USER DATA policy item does not contain any definitions for message

ID DFS810A an automation table entry for this message is not included in the automation table that is automatically built.

## IMS Automation: EVISPINM

The EVISPINM table has been retired~~ in SA z/OS 3.1~~, however, it is possible that you may have defined additional IMS or user messages in EVISPINM that are to be exposed to automation. You should review the user-modified EVISPINM from SA z/OS 2.3 and migrate any user-defined messages that still require automation to the MESSAGES/USER entry of the policy database for any affected IMS Control Regions.

## Equivalents to Retired Commands

| **Command Retired in V3.1** | **Equivalent** |
|---|---|
| CICSDLY | MDFYSHUT |
| CICSPOST | INGEVENT |
| EVEED003 | AOFCPMSG |
| EVEEMIGR | — |
| EVIED003 | AOFCPMSG |
| EVJESHUT | INGREQ RESTART=YES |
| GWTRACE | — |
| IMSPOST | INGEVENT |
| INGHC | — |
| INGPW | GETPW |
| OPCSRST | SRSTAT |
| UPDPW | — |

## Coupling Facility Structures

When you no longer have any previous releases coexisting with SA z/OS 3.1, you can remove the ING_HEALTHCHKLOG structure from the CFRM policy. SA z/OS 3.1 only requires the HSA_LOG structure (see “Step 12: Customizing the System Logger” on page 111).

## Incompatibilities

Be aware of the following incompatibilities when carrying out migration:

- The ASF command requires the DATE to be specified with a 4-digit year, for example:

  ```
  ASF  REQ=REPL,ID=resource,DATE=mm/dd/yyyy
  ```

- The following messages return the date in the `mm/dd/yyyy` format instead of `mm/dd/yy`:
  - AOF156I
  - AOF157I
  - AOF161I
- Message AOF150I (`STATISTICS DISPLAY REQUESTED FOR` *`from_resource`* `THRU` *`to_resource`*) is issued even when *from_resource* happens to equal *to_resource*.

- Message AOF151I (`ID=` *resource*, `TYPE=` *type*, `STATUS=` *status*) includes the TYPE and STATUS keywords both when issued by the ASF command and when issued by the ASFUSER command. For the ASFUSER command, however, the TYPE and STATUS fields are blank
- The trigger and schedule columns are not longer shown in the DISPSTAT command output.
- The EVIEX002 and EVIEX003 commands no longer support the following fields:
  - SERVSTARTDT
  - SERVENDDT
  - STARTOPT
  - STARTTYPE
- The CICS autotask logon feature is no longer supported.
- The AOFEXSTA exit is no longer invoked for WTOR reply handling and SDF updates.
- The following message IDs (Message/User data Policy) are reserved for SA z/OS usage:
  - VTAMDN
  - VTAMUP
- CICS VTAM ACB recovery has been removed.
- MsysOps is no longer a valid or supported subtower.

# Appendix F. IMS Automation Migration

This section describes migration steps that are required for IMS applications when upgrading to SA z/OS 3.2. It has the following subsections:

- “Removed Functions”
- “Automation of IMS Subsystems”
- “Defining IMS Actions” on page 223
- “IMS Startup” on page 224
- “IMS Shutdown” on page 225
- “Recovery of IMS Control Region” on page 230
- “Recovery of IMS Dependent Region” on page 232
- “IMS Transaction Recovery” on page 233
- “IMS Monitoring” on page 233
- “Migration of EXIEX002 Calls” on page 243
- “XRF Support” on page 245

Each section describes the behavior in SA z/OS 3.1, the changes that have been made for SA z/OS 3.2 and the migration that is required.

## Removed Functions

The following functions have been removed in SA z/OS 3.2:

- PPI communication between NetView and IMS
- The advanced automation option in the form of common global variable AOFIMSCMDMSG
- Common routine EVIEX002 no longer supports the parameters DCSTATUS, DEPREGID, ENDDT, LASTABENDCODE, RUNSTARTYPE, STARTDT, STARTTYPE and VER.

## Automation of IMS Subsystems

The following migration instructions apply only for the automation of those subsystems that have been defined as application type IMS.

Though it is some time ago that the formerly separately offered IMS feature has been integrated in the base SA z/OS product, there is still specific program code for the IMS automation and there are still IMS-specific policy items and IMS-specific needs for the automation policy definitions.

Because the base functionality of SA z/OS has been enhanced to provide most of the automation needs of the IMS automation, large parts of the IMS feature code has meanwhile become obsolete. This allows for a tighter integration of the IMS automation in the base product, provided that the automation policy definitions are adapted to the needs of the base SA z/OS functions.

SA z/OS 3.2 now exploits the base functionality for startup, shutdown, recovery and monitoring of IMS applications. Because the obsolete IMS feature code is no longer included in SA z/OS 3.2, some of the IMS-feature-specific definitions in the automation policy will no longer be applicable and therefore have to be migrated before the automation policy can be used by SA z/OS 3.2. Due to its complexity this adaptation is expected to be done manually during the upgrade to SA z/OS 3.2. No automatic conversion will be provided.

With this integration step the automation of an IMS subsystem is now covered to a large extent by the base SA z/OS functionality and the IMS-specific automation policy definitions are reduced to a minimum. This standardization reduces the complexity and facilitates the administration of system automation as well as system operation.

The migration from SA z/OS 2.3 or SA z/OS 3.1 to SA z/OS 3.2 requires the following steps:

1. Installation of the compatibility APARs on each system in the sysplex:
   - OA20402 for SA z/OS 2.3
   - OA20403 for SA z/OS 3.1
2. Conversion of the automation policy database by opening it with the SA z/OS 3.2 customization dialog
3. Step-by-step upgrade of each system in the sysplex to SA z/OS 3.2

The necessary changes in the IMS-feature-specific definitions in the automation policy for the migration to SA z/OS 3.2 do not need to be done at the same time on all systems.

Automation policy changes that only exploit functions of SA z/OS 2.3 or SA z/OS 3.1, or small function enhancements that are delivered with the appropriate compatibility APAR, can be started after installing the compatibility APAR on the relevant system.

Other automation policy changes exploit functions of SA z/OS 3.2 and therefore cannot be done before the relevant agent is running with SA z/OS 3.2. Furthermore, when functions are dropped during the migration path, the related policy definitions may possibly have to be removed at a certain time, particularly if these old definitions cannot coexist with the new definitions. If the coexistence of new and old definitions for a function is possible, the migration of this function can be done at any time after upgrading the system to SA z/OS 3.2.

The description of the migration steps in this section includes comments about the point in time or the time period during the migration path when the required changes have to be done.

The migration description refers to the following states or times:

| State or Time | Description |
|---|---|
| Pre-SA z/OS 3.2 | SA z/OS 2.3 or SA z/OS 3.1 automation agent with the corresponding compatibility APAR OA20402 or OA20403 |
| Upgrading ACF | When converting the automation policy database by opening it with the SA z/OS 3.2 customization dialog. |
| Pre-SA z/OS 3.2 with new ACF | SA z/OS 2.3 or SA z/OS 3.1 automation agent with a SA z/OS 3.2 automation policy database. |
| Upgrading agent | When upgrading the SA z/OS automation agent with the SA z/OS 3.2 run time libraries. |
| SA z/OS 3.2 | SA z/OS 3.2 automation agent with a SA z/OS 3.2 automation policy database. |

| State or Time | Description |
|---|---|
| Only SA z/OS 3.2 in sysplex | The SA z/OS automation agents on all systems in the sysplex have been upgraded to SA z/OS 3.2. |

## Defining IMS Actions

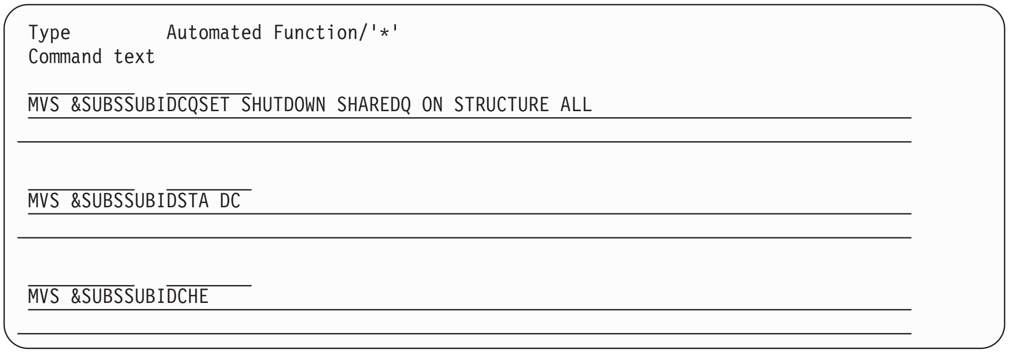
The recommended way to define automated actions in the SA z/OS automation policy that have to be executed by IMS is to specify an IMS command, prefixed with the IMS subsystem ID and to issue it with the NetView MVS command. The subsystem ID from the IMS control region itself can be retrieved from the task global variable &SUBSSUBID.

When specifying the command in the application policy object of a dependent IMS region that has a HasParent relationship to the IMS control region with sequence number 1, the command can be prefixed with the task global variable &SUBPSUBID to address it to the related IMS control region. However make sure that the IMS control region is in the first position in the list of defined parents.

If a Command Prefix has been defined in the APPLICATION INFO policy item for the IMS control region, the task global variable &SUBSCMDPFX can also be used to prefix the IMS command instead of the subsystem ID.

### Example

The poststart commands for the IMS control region can be specified as shown in the following example.



```
Type        Automated Function/'*'
Command text

MVS &SUBSSUBIDCQSET SHUTDOWN SHAREDQ ON STRUCTURE ALL


MVS &SUBSSUBIDSTA DC


MVS &SUBSSUBIDCHE
```

Figure 29. Example of IMS Poststart Commands

As soon as the compatibility APARs are applied, the task global variables &SUBSSUBID and &SUBPSUBID are available on systems running SA z/OS 2.3 and SA z/OS 3.1.

IMS actions can also be defined to be issued as the reply to the outstanding WTOR of the IMS control region. This method is only applicable to the DC control region, because the DB control region does not allow communication via an outstanding WTOR. Furthermore, when issuing a sequence of actions, processing will be throttled, because each time that a reply is issued, SA z/OS first has to wait for the next outstanding WTOR before it can continue with issuing the next reply. This method is therefore not recommended.

IMS commands can also be issued via IMSCMD. But IMSCMD acts as a stub for INGIMS for compatibility and therefore provides only an indirect method of issuing actions. Furthermore, it will be withdrawn in a future release of SA z/OS. Thus the use of IMSCMD is not recommended and should be replaced gradually.

# IMS Startup

## IMS Startup in SA z/OS 3.1

The start commands for IMS control regions are defined in the STARTUP policy item for each valid start type.

The same applies for IMS dependent regions that the **External Startup** field in the AUTOMATION INFO policy item has not been set to ALWAYS for.

Imbedded &APPLPARMS or &EHKVAR1 variables in these commands are replaced by the value that has been specified in either of the following when requesting the start with the INGREQ command:

- The APPLPARMS parameter of the INGREQ command
- The **Appl Parms** field of the INGREQ input panel

In the automation policy item MESSAGES/USER DATA for message ID CQSET, the /CQSET command is specified, which is to be issued during IMS startup to cause a Structure Checkpoint at CQS shutdown.

For message ID DFS994I, the commands required in response to the DFS994I *xxxx* START COMPLETED message are defined. This message follows the writing of a checkpoint to the IMS system log.

## Changes for IMS Startup in SA z/OS 3.2

The &EHKVAR1 variable in start commands is no longer replaced by the value of the APPLPARMS parameter or the **Appl Parms** field of the INGREQ command.

The commands that are defined for message ID CQSET, are no longer issued during IMS startup.

The command and reply definitions for message DFS994I are no longer selected by IMS-specific program code. Instead these definitions are always issued in response to a received DFS994I message.

## Migration of IMS Startup

To migrate the startup definitions in the automation policy for subsystems of type IMS, do the following:

1. Replace the &EHKVAR1 variable in startup commands for IMS control regions and IMS dependent regions with the &APPLPARMS variable.
2. Move the commands that are defined for message ID CQSET to the STARTUP policy item and specify them as POSTSTART commands.
3. Remove the command and reply definitions for message ID DFS994I and specify them as corresponding POSTSTART commands in the STARTUP policy item.

### Migration Schedule

| Activity | Pre-SA z/OS 3.2 | Upgrading ACF | Pre-SA z/OS 3.2 with new ACF | Upgrading Agent | SA z/OS 3.2 |
|---|---|---|---|---|---|
| 1. Move definitions from CQSET to POSTSTART | x | x | x | | |
| 2. In startup commands, replace &EHKVAR1 by &APPLPARMS | x | x | x | | |
| 3. Move definitions from DFS994I to POSTSTART | x | x | x | | |

# IMS Shutdown

## Shutdown of IMS Control Regions in SA z/OS 3.1

For the shutdown of the IMS control region the feature routine EVIET001 has to be specified as shutdown command in the SHUTDOWN policy item of the IMS control application.

The shutdown commands for the different shutdown phases have the following format:

```
EVIET001 subsystem shutdown_type shutdown_option
```

with:

*shutdown_type*
NORM | IMMED | FORCE

*shutdown_option*
DUMPQ | BACKUP | FREEZE | PURGE | DUMP | NODUMP

For a NORM or IMMED shutdown, the only allowed shutdown options are:

**PURGE or FREEZE** for a DBCTL region

**PURGE or FREEZE or DUMPQ** for a CTL region

For a FORCE shutdown, the only allowed shutdown options are DUMP or NODUMP.

The shutdown option BACKUP is used by the XRF function.

The shutdown options are used to select the IMS shutdown commands that have to be issued by SA z/OS to shutdown the IMS control region. The IMS shutdown commands are defined in the automation policy item MESSAGES/USER DATA for message ID SHUTTYPES with the shutdown options as selections.

The shutdown option that has been specified in the automation policy, can be overwritten when requesting the shutdown via the INGREQ command. For this purpose, the new shutdown option can be specified in the **Appl Parms** field in the format: OPTION=*shutdown_option*

Additional commands can be defined for special message IDs that are then issued by the IMS feature of SA z/OS during the shutdown process.

| Message ID | Description |
|---|---|
| BRO | Broadcast a message prior to the shutdown of a CTL region and delay issuing the SHUTINIT commands by the time interval as specified in the IMS CONTROL policy item. |
| PRECHKP | Commands to be issued before issuing the shutdown commands, as defined for message ID SHUTTYPES. The PRECHKP commands are issued prior to the HOLDQ commands.<br>Not issued when shutdown type FORCE. |
| HOLDQ | Commands to be issued after the PRECHKP commands and before issuing the shutdown commands, as defined for message ID SHUTTYPES.<br>Not issued when shutdown type FORCE. |
| SHUTTYPES | Shutdown commands for each shutdown option. The shutdown option is specified as selection.<br>In commands to selection DUMP or NODUMP, the variable $EHKVAR1 is replaced by the IMS control region job name. |
| POSTCHKP | Commands to be issued immediately after the shutdown commands as defined for message ID SHUTTYPES.<br>Not issued when shutdown type FORCE. |
| RELEASEQ | Commands to be issued after shutdown completion.<br>Not issued when shutdown is initiated with shutdown type FORCE. |
| STOPREGION | Commands to stop IMS dependent message regions.<br>With shutdown type NORM, the CANCEL commands are selected and issued with a delay of 3 minutes after having issued the shutdown and POSTCHKP commands of the IMS control region.<br>With shutdown type IMMED, the ABEND commands are selected and issued without delay after having issued the shutdown and POSTCHKP commands of the IMS control region. |
| STOPFPREGION | Commands to stop fast path regions.<br>With shutdown type NORM, the CANCEL commands are selected and issued with a delay of 3 minutes after having issued the shutdown and POSTCHKP commands of the IMS control region.<br>With shutdown type IMMED, the ABEND commands are selected and issued without delay after having issued the shutdown and POSTCHKP commands of the IMS control region. |
| STOPBMPREGION | Commands to stop batch message regions.<br>With shutdown type NORM, the CANCEL commands are selected and issued with a delay of 3 minutes after having issued the shutdown and POSTCHKP commands of the IMS control region.<br>With shutdown type IMMED, the ABEND commands are selected and issued without delay after having issued the shutdown and POSTCHKP commands of the IMS control region. |

## Changes for Shutdown of IMS Control Regions in SA z/OS 3.2

In SA z/OS 3.2, the commands to be issued during the shutdown process of the IMS control region no longer needs to be defined in an IMS-specific way, but they are defined same as for other subsystems that are processed by the base functions of SA z/OS.

&EHKVAR*n* variables in shutdown commands are no longer replaced in an IMS-feature-specific way.

The BRO message ID is no longer supported.

## Migration of IMS Control Region Shutdown

The IMS-related definitions in the automation policy database of SA z/OS 3.1 have to be adapted to definitions that are used by the base functionality of SA z/OS. This means that the shutdown commands are now specified via the automation policy item SHUTDOWN for the appropriate phase INIT, NORM, IMMED, FORCE or FINAL, dependent at which time in the shutdown process they should be issued.

When defining commands and replies, escalation processing can be implemented by exploiting pass selections.

When defining commands and replies, make use of the task global variables SUBS*xxxx* and SUBP*xxxx*, which are provided by the common routines ACFCMD and ACFREP respectively, instead of using the IMS-specific &EHKVAR*n* variables.

To migrate the shutdown definitions in the automation policy for subsystems of type IMS, move the command or reply definitions for the message IDs as listed in the left column of the following table from the MESSAGES/USER DATA policy item to the appropriate shutdown phase in the SHUTDOWN policy item.

| Message ID in SA z/OS 2.3 or SA z/OS 3.1 | Shutdown Phase in SA z/OS |
|---|---|
| PRECHKP | SHUTINIT |
| HOLDQ | SHUTINIT |
| SHUTTYPES | SHUT*xxx* [*] |
| POSTCHKP | SHUT*xxx*, SHUTFINAL |
| RELEASEQ | SHUTFINAL |

[*] Use &SUBSJOB instead of &EHKVAR1 to imbed the IMS control region job name in the defined commands.

As soon as IMS commands instead of EVIET001 are specified as shutdown commands, the definitions for the special message IDs PRECHKP, HOLDQ, SHUTTYPES, POSTCHKP and RELEASEQ are no longer used. Therefore if these definitions have been copied instead of moved, they can be removed later after upgrading the automation agent.

## Migration Schedule

| Activity | Pre-SA z/OS 3.2 | Upgrading ACF | Pre-SA z/OS 3.2 with new ACF | Upgrading Agent | SA z/OS 3.2 | Only SA z/OS 3.2 in sysplex |
|---|---|---|---|---|---|---|
| Move definitions from PRECHKP to SHUTINIT<br>Move definitions from HOLDQ to SHUTINIT<br>Define the SHUT*xxx* commands<br>Move definitions from POSTCHKP to SHUT*xxx* or SHUTFINAL<br>Move definitions from RELEASEQ to SHUTFINAL | x | x | x | | | |
| If previous definitions have been copied: Remove definitions for PRECHKP, HOLDQ, SHUTTYPES, POSTCHKP and RELEASEQ after having replaced EVIET001 by IMS commands. | x | x | x | x | x | |
| Cleanup the command and reply definitions for message ID BRO | | | | | | x |
| **Note:** The definitions for the special message IDs STOPREGION, STOPFPREGION, and STOPBMPREGION cannot be removed before the shutdown definitions for the dependent IMS regions have been migrated. | | | | | | |

# Shutdown of other IMS Region Types in SA z/OS 3.1

If the **External Shutdown** field of the AUTOMATION INFO policy item is not set to ALWAYS, automation policy definitions for the shutdown of any IMS region type other than the control region exists as follows.

A HASPARENT relationship is defined between the dependent region and the owning IMS control region.

For the shutdown of the IMS dependent region the feature routine EVIET00J is specified as shutdown command in the SHUTDOWN policy item of the IMS dependent application.

The shutdown commands for the different shutdown phases have the following format:

```
EVIET00J &SUBSAPPL shutdown_type
```

with:

| | |
|---|---|
| *shutdown_type* | NORM \| IMMED \| FORCE |

The passed shutdown type and the region type are used to select the IMS shutdown commands that have to be issued by SA z/OS to shutdown the IMS dependent region. The IMS shutdown commands are defined in the automation policy item MESSAGES/USER DATA of the associated IMS control region for a message ID that relates to the region type as follows:

| Region type | Associated Message ID |
|---|---|
| BMP regions | STOPBMPREGION |
| FP regions | STOPFPREGION |
| Dependent message regions | STOPREGION |

The shutdown type is used to select the commands to be issued:

| Shutdown Type | Selection |
|---|---|
| NORM | NORMAL |
| IMMED | ABEND |
| FORCE | CANCEL |

Because defined replies are issued via the outstanding WTOR of the relating IMS control region, the shutdown processing of the dependent regions is serialized on the work operator of the IMS control region.

## Changes for Shutdown of other IMS Region Types in SA z/OS 3.2

In SA z/OS 3.2, the commands to be issued during the shutdown process of the IMS dependent region no longer needs to be defined in an IMS-specific way, but they are defined same as for other subsystems that are processed by the base functions of SA z/OS. The shutdown of the IMS dependent regions is now processed parallel by the responsible work operators for the dependent regions.

With the exception of &EHKVAR1, &EHKVAR*n* variables in shutdown commands are no longer replaced in an IMS-feature-specific way.

### Migration of other IMS Region Type Shutdown

The IMS-related definitions in the automation policy database of SA z/OS 3.1 have to be manually adapted to definitions that are used by the base functionality of SA z/OS. This means that the shutdown commands are now to be specified via the automation policy item SHUTDOWN of the IMS dependent region for the appropriate phase NORM, IMMED or FORCE. Keep in mind that IMS dependent regions have no outstanding WTOR. Therefore it does not make sense to define replies as shutdown actions for IMS dependent regions.

When defining commands, make use of the task global variables SUBS*xxxx* and SUBP*xxxx* that are provided by common routine ACFCMD. The Transaction ID is provided by variable &EHKVAR1 that can also be imbedded in the shutdown commands. Other IMS-specific &EHKVAR*n* variables are no longer supported.

As soon as IMS commands instead of EVIET00J are specified as shutdown commands, the definitions for the special message IDs STOPREGION, STOPFPREGION and STOPBMPREGION in the MESSAGES/USER DATA policy item of the IMS control region are no longer used.

**Note:** An automation agent based on SA z/OS 3.2 is still able to shut down IMS dependent regions with the shutdown definitions of SA z/OS 3.1. Therefore the shutdown definitions for IMS dependent regions do not necessarily have to be migrated before changing to SA z/OS 3.2.

## Migration Schedule

| Activity | Pre-SA z/OS 3.2 | Upgrading ACF | Pre-SA z/OS 3.2 with new ACF | Upgrading Agent | SA z/OS 3.2 |
|---|---|---|---|---|---|
| Define the SHUT*xxx* commands for the dependent regions | x | x | x | x | x |
| Remove definitions for STOPREGION, STOPFPREGION and STOPBMPREGION after having replaced EVIET001 and EVIET00J by IMS commands. | x | x | x | x | x |

# Recovery of IMS Control Region

## Recovery of IMS Control Region in SA z/OS 3.1

The IMS feature code reacts to the messages DFS627I, DFS629I, IEF450I and IEF743I that are issued in case of an abend of the IMS control region.

Each of these messages triggers an IMS feature routine via automation table entries:

| Message ID | REXX Routine | AT Entry Type |
|---|---|---|
| DFS629I | EVIER000 | Forced |
| DFS627I | EVIER001 | Forced |
| IEF450I | EVIER002 via TERMMSG | Forced |
| IEF743I | EVIER002 via TERMMSG | Recommended |

### ABCODES

Routine EVIER000 extracts the system or user code from the triggering message DFS629I and uses it for a check with codes defined for message ID ABCODES to get back ABENDING or STOPPING as new status for the application.

The code definitions are expected in the format:

```
Code 1          Code 2          Code 3          Value Returned
IMSuuuu                                         ABENDING
SYSsss                                          STOPPING
```

STOPPING as specified Value Returned only prevents the abended subsystem to be restarted, if the restart option is not ALWAYS.

## Changes for Recovery of IMS Control Region in SA z/OS 3.2

In SA z/OS 3.2, no IMS feature code is involved in processing the recovery of IMS control regions.

When one of the messages DFS629I, IEF450I or IEF743I is issued, TERMMSG is called via the automation table.

For message DFS629I, which reports a system or user abend for an IMS control region, the system or user abend code is extracted from the message text and passed as CODE*x* parameters to TERMMSG.

In TERMMSG, the passed codes are used to check codes that are defined for message DFS629I for the issuing application or for MVSESA (value of common global variable AOFSYSTEM). The value returned of the matching codes determines the new status of the application.

The automation table entry for message DFS629I is no longer forced but recommended.

The automation table entry for message DFS627I has been removed.

&EHKVAR*n* variables in defined commands are no longer replaced in an IMS-feature-specific way.

## Migration of IMS Control Region Recovery

Check the defined commands for message DFS989I. Remove an imbedded &EHKVAR7 variable by specifying the commands as recommended in “Defining IMS Actions” on page 223.

The abend code definitions for message ID ABCODES that is specified in the MESSAGES/USER DATA policy item of the IMS control region have to be moved to message DFS629I for the issuing application. Whenever possible, the code definitions should be made at a class level.

The extracted values from the DFS629I message that are checked for matching with the defined codes are as follows. Note the changed syntax for the code values, which is now consistent with that of message IEF450I.

**Code1** Job name

**Code2** System abend code in the format S*sss* or S000 if not present.

**Code3** User abend code in the format U*uuuu* or U0000 if not present.

The value returned determines the new status for the application.

### Example

| Code 1 | Code 2 | Code 3 | Value Returned |
|---|---|---|---|
| * | S000 | U0020 | ABENDING |
| * | S000 | U0075 | BREAKING |
| * | S000 | U0113 | ABENDING |
| * | S000 | U0707 | ABENDING |
| * | * | * | BREAKING |

**Note:** The changed behavior in the recovery processing is provided by generic routine TERMMSG of SA z/OS 3.2. Therefore the recovery processing just changes to the new behavior at the moment, when the automation agent is upgraded to SA z/OS 3.2. Until that moment, message DFS629I still has to trigger the IMS feature routine and not TERMMSG.

But when using the automation table that is built automatically by the policy build process, the automation table statement for DFS629I changes at the moment, when upgrading the ACF to SA z/OS 3.2. To prevent the usage of this changed statement, an additional automation table is provided with the original, unchanged statement. This compatibility automation table, INGMSG32, should be activated from the moment the upgraded ACF is used as long as the automation agent is not yet running on SA z/OS 3.2. To do this, specify INGMSG32 in front of INGMSG01 as Automation Tables in the SYSTEM INFO policy item of the System policy object via the customization dialog and remove it when the automation agent is upgraded to SA z/OS 3.2.

The code definitions for DFS629I can be prepared but are not used as long as the automation agent does not run on SA z/OS 3.2.

**Migration Schedule:**

| Activity | Pre-SA z/OS 3.2 | Upgrading ACF | Pre-SA z/OS 3.2 with new ACF | Upgrading Agent | SA z/OS 3.2 |
|---|---|---|---|---|---|
| Replace &EHKVAR7 in defined commands for message DFS989I | x | x | x | x | |
| Convert ABCODES definitions to DFS629I | Prepare | Prepare | Prepare | x | |
| If using AT built by SA z/OS 3.2, include compatibility AT INGMSG32 | | x | | | |
| If using AT built by SA z/OS 3.2, remove compatibility AT INGMSG32 | | | | x | |
| Clean up the code definitions to ABCODES | | | | | x |

## Recovery of IMS Dependent Region

### Changes for Recovery of IMS Dependent Region in SA z/OS 3.2

In SA z/OS 3.2, code definitions for message ID TPABEND in the MESSAGES/USER DATA policy item are no longer used.

### Migration of IMS Dependent Region Recovery

Code definitions for message ID TPABEND can be removed as soon as all systems in the sysplex have been upgraded to SA z/OS 3.2.

#### Migration Schedule

| Activity | Pre-SA z/OS 3.2 | Upgrading ACF | Pre-SA z/OS 3.2 with new ACF | Upgrading Agent | SA z/OS 3.2 | Only SA z/OS 3.2 in sysplex |
|---|---|---|---|---|---|---|
| Remove code definitions for message ID TPABEND | | | | | | x |

## IMS Transaction Recovery

### Changes for IMS Transaction Recovery in SA z/OS 3.2

In SA z/OS 3.2 the transaction recovery processing has been restructured, so that the state/action table is no longer used.

### Migration of IMS Transaction Recovery

After upgrading the automation agent to SA z/OS 3.2, the link to the IMS state/action table EVISS005 can be removed via the STATE ACTION TABLE policy item.

As soon as all systems in the sysplex have been upgraded to SA z/OS 3.2, the state/action table EVISS005 can be removed via the Product Automation policy object.

#### Migration Schedule

| Activity | Pre-SA z/OS 3.2 | Upgrading ACF | Pre-SA z/OS 3.2 with new ACF | Upgrading Agent | SA z/OS 3.2 | Only SA z/OS 3.2 in sysplex |
|---|---|---|---|---|---|---|
| Remove link to state/action table EVISS005 | | | | x | x | x |
| Remove state/action table EVISS005 | | | | | | x |

## IMS Monitoring

The IMS feature code in SA z/OS 3.1 includes the following monitoring functions:

- Monitoring of online log data sets (OLDS)
- Monitoring of recovery control data sets (RECON)
- Monitoring of VTAM ACB
- Monitoring of Multiple Systems Coupling (MSC) links between IMS systems

In SA z/OS 3.2 the IMS monitoring functions are redesigned to be implemented by exploiting the concept of monitor resources, which is part of the base functionality of SA z/OS. This concept allows:

- Starting and stopping the monitoring function
- Implementing a reasoned dependency management that keeps the monitoring function only up and running as long as the required resources are available
- Defining health status commands that are issued when the health status changes

### Enhancements in Monitor Resources

To be able, to implement the IMS monitoring functions based on the concept of monitor resources, this concept has been extended in SA z/OS 3.2 by the following two features:

- Specifying object and job name when defining a monitor resource to be able to map a monitor event to the relating monitor resource.
- Specifying “Check” as status when defining a message as status message with action AUTO in the MESSAGES/USER DATA policy item of a monitor resource. Such a status message is defined to update the health status of a monitor

resource. Selecting "Check" as health status means that the health status is first to be evaluated with the defined monitor command at the moment, when the status message is received, before the health status of the monitor resource can be updated.

## Operational Changes in SDF and NMC

Removing the monitor functions of OLDS, RECONS and MCS links as having been implemented by the IMS feature code and replacing it by the concept of monitor resources results in a change of the status display in SDF, because the resulting status of OLDS, RECON and MSC links are no longer reflected in the color of the relating objects in the IMS MONITOR PANEL. But the monitor status is now reflected in the color of the listed monitor resources and hence in the color of the relating IMS application caused by a degraded health status. In case of a degraded monitor status the operator no longer needs to analyze which IMS application is affected by the detected problems.

## Optional Monitor Features

All these monitor functions are optional. They only need to be implemented, if the status of these objects should be monitored.

## OLDS Monitoring in SA z/OS 3.1

The IMS feature in SA z/OS 3.1 includes an OLDS monitoring function that monitors the status of the online log data sets (OLDS) of an IMS control region and takes recovery actions if needed.

For this purpose, the IMS command DISPLAY OLDS is issued to analyze the status of the listed OLDS data sets. If needed, OLDS data sets are started or stopped. Hereby, OLDS data sets that have been defined as spare OLDS data sets are started at last and stopped at first. The recovery actions are controlled via message automation.

### Automation Settings in Automation Policy

The automation settings for OLDS monitoring are provided as user data for the special message ID OLDS that is defined in the MESSAGES/USER DATA policy item of the IMS control region with the following keyword-data pairs:

| Keyword | Data | Description |
|---|---|---|
| MINIMUM | nn | The minimum number of OLDS that must be available at all times.<br>Default: 50% of the normal number of OLDS, where the normal number of online data sets is determined by counting the listed OLDS in the response of the DISPLAY OLDS command. |
| SPARES | (nn,nn...) | The spares are online data sets that IMS automation activates when the number of available online data sets drops below the minimum. |
| ARCHIVETIME | hh:mm:ss | The archive time is the maximum length of time archive jobs take to run. |

| Keyword | Data | Description |
|---|---|---|
| RETRYCNT | n | The retry count is the number of times that IMS automation will attempt to acquire an outstanding reply ID when activating or deactivating a spare OLDS.<br>Default: 5 retries |
| BACKOUT | nn | The maximum number of OLDS that can have an OTHER-STS of BACKOUT. |

### Processing the Monitor Function

The OLDS monitoring function is based on message automation via the automation table.

Appropriate messages are used to update the state action table EVISS003 and to update the status of the OLDS object in SDF and NMC.

The resulting recovery processing depends on:

- The automation flag for resource *subsystem*.OLDS
- Defined threshold levels for resource *subsystem*.OLDS.

## Changes for OLDS Monitoring in SA z/OS 3.2

The OLDS monitoring function is no longer part of the IMS feature code, but is assumed by the base SA z/OS functions.

### Automation Settings in Automation Policy

The input parameters for the OLDS monitoring are further on defined as user data for the special message ID OLDS, but are reduced to the following keyword-data pairs:

| Keyword | Data | Description |
|---|---|---|
| MINIMUM | nn | The minimum number of OLDS that must be available at all times.<br>Default: 50% of the normal number of OLDS, where the normal number of OLDS is determined by counting the listed OLDS in the response of the DISPLAY OLDS command. |
| SPARES | (nn,nn...) | The spares are online data sets that IMS automation activates when the number of available online data sets drops below the minimum. |
| BACKOUT | nn | The maximum number of OLDS that can have an OTHER-STS of BACKOUT. |

### Monitor Resources

To maintain the full OLDS monitoring function in SA z/OS 3.2, two monitor resources have to be defined: one to monitor the number of available OLDS, the other to monitor for excessive switching of the OLDS. The health status of the monitored object is propagated to the health status of the application via a HASMONITOR relationship between the IMS control region and each monitor. Additional HASPASSIVEPARENT and FORCEDOWN/WhenObservedDown relationships between each monitor resource and the IMS application group mean that the monitor resources are only active during the UP time of the IMS application group. This ensures that the monitor resources can rely on all required functions.

The two monitor resources have to be defined with the following characteristics.

**IMS OLDS Monitor:**

| | |
|---|---|
| **Monitored Object** | OLDS |
| **Monitored Jobname** | Job name of the IMS control region |
| **Monitor command** | INGRMIOL |
| **Monitoring Interval (optional)** | *hh*:*mm* |

Without specified monitoring interval, the monitor command is at least issued initially when the monitor resource is started.

Depending on the monitor results, INGRMIOL ends with the following return codes that are mapped to the relating health status for the monitor resource.

| Return Code | Health Status | Description |
|---|---|---|
| 1 | BROKEN | Monitor encountered a severe error |
| 2 | FAILED | DISPLAY OLDS failed |
| 3 | NORMAL | No problem found by OLDS monitoring |
| 4 | WARNING | Needed to start spare OLDS to have the minimum in AVAILABLE status<br>or<br>AUTOMATIC ARCHIVE is off |
| 5 | MINOR | Could not start enough spare OLDS to have the minimum in AVAILABLE status |
| 6 | CRITICAL | Number of OLDS in status BACKOUT exceeds maximum limit |

Primarily besides this optional active monitoring with the INGRMIOL monitor command after each monitoring interval, passive monitoring is used to update the health status of the monitor resource. The following IMS messages are used to trigger health status updates:

```
DFS3256I OPEN/ALLOCATION FAILED ON ddname
DFS3257I ONLINE LOG NOW SWITCHED - FROM DFSOLPxxx TO ddname2
DFS3258A LAST ONLINE LOG DATA SET IS BEING USED - NEED ARCHIVE
DFS3260I ONLINE LOG DATA SET SHORTAGE - NEED ANOTHER DATA SET
```

Because these messages only indicate OLDS-related activities that may or may not change the health status of the monitor resource, the defined monitor command first has to be issued to analyze the actual situation and to evaluate the health status, before it can be updated. To achieve this, these messages have to be defined as status messages via the MESSAGES/USER DATA policy item of the monitor resource, with the status `Check`.

**IMS OLDS Switch Frequency Monitor:**

| | |
|---|---|
| **Monitored Object** | OLDS_SWITCH |

**Monitored Jobname** Job name of the IMS control region

**Monitor command** (None.)

The switch frequency is determined by passive monitoring of message DFS3257I.

For this, the following commands have to be defined for message DFS3257I in the policy item MESSAGES/USER DATA of the IMS control region.

```
Pass/Selection Automated Function/'*'
Command Text
 INFR
 INGMON OLDS_SWITCH,JOBNAME=&SUBSJOB,STATUS=WARNING,INFO=(MSG,INFREQUENT THRESHO
LDS LIMIT REACHED FOR OLDS SWITCHING)
 FREQ
 INGMON OLDS_SWITCH,JOBNAME=&SUBSJOB,STATUS=MINOR,INFO=(MSG,FREQUENT OLDS SWITCH
ING DETECTED)
 CRIT
 INGMON OLDS_SWITCH,JOBNAME=&SUBSJOB,STATUS=CRITICAL,INFO=(MSG,CRITICAL OLDS SWI
TCHING FREQUENCY REACHED)
 ALWAYS
 INGMON OLDS_SWITCH,JOBNAME=&SUBSJOB,STATUS=NORMAL,INFO=(MSG,OLDS SWITCHING FREQ
UENCY IS NORMAL)
```

**OLDS Status Display:** The status of OLDS is no longer displayed in the IMS Monitor Panel in SDF and on NMC but is reflected in the color of the displayed monitor resources and hence in the color of the relating IMS application.

## Migration of OLDS Monitoring

Because the newly designed OLDS monitoring function exploits functional enhancements of SA z/OS 3.2, it cannot be activated before upgrading the automation agent to SA z/OS 3.2.

After upgrading the automation agent to SA z/OS 3.2 on a system:

1. Define the two monitor resources for OLDS monitoring as described above.
2. Define messages DFS3256I, DFS3257I, DFS3258A and DFS3260I as status messages with health status “Check” in the MESSAGES/USER DATA policy item of the monitor resource for the OLDS object.
3. Define commands for selections ALWAYS, INFR, FREQ and CRIT under the message ID DFS3257I in the MESSAGES/USER DATA policy item of the IMS control region.
4. Check the automation flag settings for resource *subsystem*.OLDS. No changes are needed due to the migration.
5. Check the defined thresholds for resource *subsystem*.DFS3257I. No changes are needed due to the migration.
6. Remove the link to the IMS state/action table EVISS003 via the STATE ACTION TABLE policy item, because this table is no longer used for OLDS monitoring.
7. Inspect the defined commands for message DFS3258A. These definitions are no longer selected by IMS-specific program code, therefore the selection names SYSTEM or LAST are no longer respected. If still needed, specify the commands as expected by the base message automation function of SA z/OS, otherwise remove the definitions.

After upgrading all systems in the sysplex to SA z/OS 3.2:

1. Remove the IMS state/action table EVISS003 via the Product Automation policy object.

2. Remove the unneeded user data definitions for the keywords ARCHIVETIME and RETRYCNT of the special message ID OLDS in the MESSAGES/USER DATA policy item of the IMS control region.

To keep the old monitoring function up and running even with the automation table that is built automatically by the policy build process of SA z/OS 3.2, an additional automation table is provided with the original, unchanged statements for the monitoring function of SA z/OS 3.1. This compatibility automation table, INGMSG32, should be activated from the moment the upgraded ACF is used as long as the automation agent is not yet running on SA z/OS 3.2. To do this, specify INGMSG32 in front of INGMSG01 as automation table in the SYSTEM INFO policy item of the System policy object via the customization dialog and remove it when the automation agent is upgraded to SA z/OS 3.2.

### Migration Schedule

| Activity | Pre-SA z/OS 3.2 | Upgrading ACF | Pre-SA z/OS 3.2 with new ACF | Upgrading Agent | SA z/OS 3.2 | Only SA z/OS 3.2 in sysplex |
|---|---|---|---|---|---|---|
| If using AT built by SA z/OS 3.2, include compatibility AT INGMSG32 | | x | | | | |
| Define the two OLDS monitor resources | | | | x | | |
| Define the status messages for the OLDS monitor resource | | | | x | | |
| Define the commands for message DFS3257I in the IMS control region entry | | | | x | | |
| Check the automation flag settings and defined thresholds for resource *subsystem*.OLDS | | | | x | | |
| If using AT built by SA z/OS 3.2, remove compatibility AT INGMSG32 | | | | x | | |
| Remove the link to state/action table EVISS003 | | | | x | x | x |
| Remove the state/action table EVISS003 | | | | | | x |
| Clean up the user data definitions for ARCHIVETIME and RETRYCNT from message OLDS | | | | | | x |

## RECON Monitoring in SA z/OS 3.1

The IMS feature in SA z/OS 3.1 includes a RECON monitoring function that checks the status of the recovery control data sets (RECON) of an IMS control region. For this purpose, the IMS command RMLIST DBRC='RECON STATUS' is issued regularly to analyze the status of the listed RECON data sets in the response to this command.

### Automation Settings in Automation Policy

The automation settings for the RECON monitoring are provided as user data for the special message ID RECONS that is defined in the MESSAGES/USER DATA policy item of the IMS control region with the following keyword-data pairs:

| Keyword | Data | Description |
|---|---|---|
| MONITOR | hh:mm:ss | Determines how often the RECONs are checked to make sure a spare is available. |
| RETRY | nn | Determines how many times RECON monitoring will recheck before informing about an UNAVAILABLE spare RECONS data set. |
| DELAY | nn | Determines the time interval between the retries. |

## Changes for RECON Monitoring in SA z/OS 3.2

The RECON monitoring function is no longer part of the IMS feature code, but is assumed by the base SA z/OS functions.

### Automation Settings in Automation Policy

RECON monitoring in SA z/OS 3.2 no longer needs any defined user data as input parameters.

### Monitor Resource

To maintain the full RECON monitoring function in SA z/OS 3.2, a monitor resource has to be defined to monitor the number of available RECON data sets. The health status of the monitored object is propagated to the health status of the IMS application via a HASMONITOR relationship between the IMS control region and the monitor resource. Additional HASPASSIVEPARENT and FORCEDOWN/WhenObservedDown relationships between the monitor resource and the IMS application group mean that the monitor resource is only active during the UP time of the IMS application group. This ensures that the monitor resource can rely on all required functions.

The monitor resource has to be defined with the following characteristics.

**IMS RECON Monitor:**

| | |
|---|---|
| **Monitored Object** | RECON |
| **Monitored Jobname** | Job name of the IMS control region |
| **Monitor command** | INGRMIRE |
| **Monitoring Interval** | *hh*:*mm* |

The monitor command is executed after each monitoring interval. It issues the following command and analyzes the status of the listed RECON data sets in the response to it:

```
RMLIST DBRC='RECON STATUS'
```

Depending on the monitor results, INGRMIRE ends with the following return codes that are mapped to the relating health status for the monitor resource.

| Return Code | Health Status | Description |
|---|---|---|
| 1 | BROKEN | Monitor encountered a severe error |
| 2 | FAILED | RMLIST DBRC='RECON STATUS' failed |
| 3 | NORMAL | No problem found by RECON monitoring |

| Return Code | Health Status | Description |
|---|---|---|
| 4 | WARNING | No COPY2 found for RECON |
| 5 | MINOR | No SPARE found for RECON |
| 6 | CRITICAL | Neither COPY2 nor SPARE found for RECON |
| 7 | FATAL | No RECON data sets found |

In addition to the active monitoring with the INGRMIRE monitor command after each monitoring interval, passive monitoring is used to update the health status of the monitor resource. The following IMS message is used to trigger health status update:

```
DSP0038I RECON INCONSISTENCY RECON HEADER RECORD NOT FOUND
```

When message DSP0038I is received, the defined monitor command has first to be issued to analyze the actual situation and to evaluate the health status, before it can be updated. For this purpose, this message has to be defined as status message via the MESSAGES/USER DATA policy item of the monitor resource, with the status `Check`.

**RECON Status Display:** The status of RECONs is no longer displayed in the IMS Monitor Panel in SDF and on NMC but is reflected in the color of the displayed monitor resource and hence in the color of the relating IMS application.

## Migration of RECON Monitoring

Because the newly designed RECON monitoring function exploits functional enhancements of SA z/OS 3.2, it cannot be activated before upgrading the automation agent to SA z/OS 3.2.

After upgrading the automation agent to SA z/OS 3.2 on a system:

- Define the monitor resource for RECON monitoring as described above.
- Define message DSP0038I as status messages with health status “Check” in the MESSAGES/USER DATA policy item of the monitor resource for the RECON object.

After upgrading all systems in the sysplex to SA z/OS 3.2:

- Remove the user data definitions for the special message ID RECONS.

### Migration Schedule

| Activity | Pre-SA z/OS 3.2 | Upgrading ACF | Pre-SA z/OS 3.2 with new ACF | Upgrading Agent | SA z/OS 3.2 | Only SA z/OS 3.2 in sysplex |
|---|---|---|---|---|---|---|
| Define the RECON monitor resource | | | | x | | |
| Cleanup the user data definitions from message RECONS | | | | | | x |

## DC Monitoring in SA z/OS 3.1

The IMS feature in SA z/OS 3.1 includes a DC monitoring function that checks the status of the VTAM ACB and the enablement of logons. For this purpose, the IMS command DISPLAY ACTIVE DC is issued once after IMS startup and the response to this command is analyzed. In case the status is not satisfied, an event is sent to SDF and NMC.

## Changes for DC Monitoring in SA z/OS 3.2

The monitoring function is no longer part of the IMS feature code, but is assumed by the base SA z/OS functions.

### Monitor Resource

To maintain the full DC monitoring function in SA z/OS 3.2, a monitor resource has to be defined to monitor the VTAM ACB status and the enablement of logons. The monitor resource is only useful for an IMS control region, but not for a DB control region. The health status of the monitored resource is propagated to the status of the IMS application via a HASMONITOR relationship between the IMS control region and the monitor resource. Additional HASPASSIVEPARENT and FORCEDOWN/WhenObservedDown relationships between the monitor resource and the IMS control region mean that the monitor resource is only active during the UP time of the IMS control region.

The monitor resource has to be defined with the following characteristics.

**IMS DC Monitor:**

**Monitored Object**
DC

**Monitored Jobname**
Job name of the IMS control region

**Monitor command**
INGRMIDC

**Monitoring Interval (optional)**
*hh*:*mm*

The monitor command is executed after each monitoring interval. It issues the following IMS command and analyses the status of the VTAM ACB and the LOGONS enablement:

```
DISPLAY ACTIVE DC
```

If no monitoring interval is specified the defined monitor command is only issued once, initially after having started the monitor resource.

Depending on the monitor results, INGRMIDC ends with the following return codes that are mapped to the relating health status for the monitor resource.

| Return Code | Health Status | Description |
|---|---|---|
| 1 | BROKEN | Monitor encountered a severe error |
| 2 | FAILED | DISPLAY ACTIVE DC failed |
| 3 | NORMAL | VTAM ACB is OPEN and LOGONS enabled |
| 4 | WARNING | LOGONS are not enabled |

In addition to the active monitoring with the INGRMIDC monitor command after each monitoring interval, passive monitoring is used to update the health status of the monitor resource. The following IMS message is used to trigger health status update:

```
DFS2111I VTAM ACB CLOSED.
```

When message DFS2111I is received, the health status should be set to WARNING. For this purpose, this message has to be defined as status message via the MESSAGES/USER DATA policy item of the monitor resource, with the status `WARNING`.

## Migration of DC Monitoring

Because the newly designed DC monitoring function exploits functional enhancements of SA z/OS 3.2, it cannot be activated before upgrading the automation agent to SA z/OS 3.2.

After upgrading the automation agent to SA z/OS 3.2 on a system:

- Define a monitor resource for DC monitoring as described above.

### Migration Schedule

| Activity | Pre-SA z/OS 3.2 | Upgrading ACF | Pre-SA z/OS 3.2 with new ACF | Upgrading Agent | SA z/OS 3.2 |
|---|---|---|---|---|---|
| Define the DC monitor resource | | | | x | |

## MSC Link Monitoring in SA z/OS 3.1

The IMS feature in SA z/OS 3.1 includes an MSC monitoring function which monitors the status of MSC links by passive monitoring and triggers recovery actions based on received messages.

Messages DFS2140, DFS2142, DFS2160I, DFS2161I, DFS2168I, DFS2169I and DFS2236 are used to update the IMS state/action table EVISS002 and to update the status of the MSC links in SDF and NMC.

In addition, messages DFS2142, DFS2161I and DFS2169I are used to trigger recovery actions that are defined in MESSAGES/USER DATA policy item of the IMS control region for the relating message ID. In these defined actions, variables &EHKVAR1 and &EHKVAR2 are replaced by the logical link number and the logical link path name.

The defined recovery actions are only issued if:

- The automation flag for minor resource *subsystem*.MSC.*link_id* is not switched off
- The frequency of the automated message does not exceed the frequent threshold level of minor resource *subsystem*.MSC.*link_id*.

## Changes for MSC Link Monitoring in SA z/OS 3.2

The MSC link monitoring function has been removed without replacement.

The base SA z/OS functions can be used instead with the following restrictions or changes:

- When defining recovery actions for received messages, variables &EHKVAR*x* can be imbedded which are replaced by the appropriate token of the triggering messages.
- The automation flags of minor resource *subsystem.msgid* are checked instead of *subsystem*.MSC.*link_id*.
- The defined thresholds are checked for minor resource *subsystem.msgid* instead of *subsystem*.MSC.*link_id*.

- The status of the MSC links is no longer displayed in the IMS Monitor Panel in SDF and on NMC

## Migration of MSC Link Monitoring

Because the newly designed MSC link monitoring function exploits functional enhancements of SA z/OS 3.2, it cannot be activated before upgrading the automation agent to SA z/OS 3.2.

After upgrading the automation agent to SA z/OS 3.2 on a system:

- In defined actions for messages DFS2142, DFS2161I and DFS2169I, imbed variable parts of the triggering message by referencing the appropriate token via the correlating &EHKVAR*x* variable. Also check the selection names for these actions.
- Replace MSC related automation flag definitions by appropriate message related definitions with the MINOR RESOURCE FLAGS policy item.
- Replace MSC related threshold definitions by appropriate message related definitions with the MINOR RESOURCE THRESH policy item.
- Remove the link to the IMS state/action table EVISS002 via the STATE ACTION TABLE policy item, because this table is no longer used for MSC link monitoring.

After upgrading all systems in the sysplex to SA z/OS 3.2:

- Remove the IMS state/action table EVISS002 via the Product Automation policy object.

### Migration Schedule

| Activity | Pre-SA z/OS 3.2 | Upgrading ACF | Pre-SA z/OS 3.2 with new ACF | Upgrading Agent | SA z/OS 3.2 | Only SA z/OS 3.2 in sysplex |
|---|---|---|---|---|---|---|
| Implement new MSC link monitoring by message automation | | | | x | | |
| Remove the link to the state/action table EVISS002 | | | | x | x | x |
| Remove the state/action table EVISS002 | | | | | | x |

## Migration of EXIEX002 Calls

Use the following methods to request information that was previously provided by the EVIEX002 common routine:

**DCSTATUS**

If a monitor resource for DC monitoring is implemented as described in “Changes for DC Monitoring in SA z/OS 3.2” on page 241, the health status of this monitor resource indicates the DC status.

The health status of the monitor resource is provided in the variable DCSTATUS when calling the following command:

```
PIPE NETV INGDATA resourcename | EDIT 224.10 1 | VAR DCSTATUS
```

where *resourcename* is the name of the monitor resource.

**ENDDT**

The ASF command can be used to obtain information about the termination time of a resource.

The following command provides date and time when the stop order for a resource was received from the automation manager. The information is delivered in the variable STOPDT.

```
PIPE NETV ASF REQ=DISPLAY ID=resource| SEP | LOC 1.26 /AOF153I LAST STOP
  EVENT: / | EDIT WORD 6.2 | VAR STOPDT
```

The following command provides date and time when the resource became unavailable, as determined by the automation manager. The information is delivered in the variable ENDDT.

```
PIPE NETV ASF REQ=DISPLAY ID=resource| SEP | LOC 1.26 /AOF153I LAST DOWN
  EVENT: / | EDIT WORD 6.2 | VAR ENDDT
```

**LASTABENDCODE**

The ASF command can be used to obtain information about the last termination code.

The following command provides the last termination code in the variable LASTABENDCODE:

```
PIPE NETV ASF REQ=DISPLAY ID=resource| SEP | LOC 1.8 /AOF154I / | EDIT
  WORD 4.1 1 | VAR LASTABENDCODE
```

**RUNSTARTYPE**

The ASF command can be used to obtain the last used start type of a resource.

The following command provides the last used start type in the variable RUNSTARTYPE:

```
PIPE NETV ASF REQ=DISPLAY ID=resource| SEP | LOC 1.26 /LAST START EVENT:
  / | EDIT WORD 10.1 1 | VAR RUNSTARTYPE
```

**STARTDT**

The ASF command can be used to obtain information about the start time of a resource.

The following command provides date and time when the start order for a resource was received from the automation manager. The information is delivered in the variable STARTDT.

```
PIPE NETV ASF REQ=DISPLAY ID=resource| SEP | LOC 1.26 /AOF153I LAST START
  EVENT: / | EDIT WORD 6.2 | VAR STARTDT
```

The following command provides date and time when the resource became unavailable, as determined by the automation manager. The information is delivered in the variable AVAILDT.

```
PIPE NETV ASF REQ=DISPLAY ID=resource| SEP | LOC 1.26 /AOF153I LAST AVAIL
  EVENT: / | EDIT WORD 6.2 | VAR AVAILDT
```

**STARTTYPE**

The start type to be used to start a resource can be requested with the INGDATA command.

The following command provides the start type in the variable STARTTYPE.

```
PIPE NETV INGDATA resource | EDIT 124.10 1 | VAR STARTTYPE
```

## XRF Support

The following migration steps are only needed if IMS control regions are executing as an XRF complex, and these IMS control regions are defined with the **XRF enabled** field set to YES in the IMS CONTROL policy item of SA z/OS.

In SA z/OS 3.2 the automation table statements for the XRF-related automation of IMS control regions have been outsourced to a separate automation table. To continue to use XRF support in SA z/OS 3.2, this automation table, INGMSGIX, has to be specified ahead of INGMSG01 as Automation Tables in the SYSTEM INFO policy item of the System policy object via the customization dialog.

The shutdown commands for the IMS control region with the **XRF enabled** field set to YES must be prefixed with `EVIRXRF STOP,`*`xrf_status`* where *xrf_status* is either ACTIVE or BACKUP. This applies to all passes.

The following command has to be defined as the POSTSTART command in the STARTUP policy item of the IMS control region:

```
EVIRXRF EVENT,UP
```

The following command has to be defined as the FINAL command in the SHUTDOWN policy item of the IMS control region:

```
EVIRXRF EVENT,DOWN
```

All migration for XRF must be done when upgrading to SA z/OS 3.2. The affected policy entries cannot be used with any earlier releases of SA z/OS.

# Appendix G. Syntax for HSAPRM00

**Notes:**

1. A sample member called HSAPRM00 is provided in the SINGSAMP sample library.
2. Records starting with a ′*′ in column 1 are treated as comments. Each parameter must be specified on a single line. Trailing comments are not supported.

```
BLOCKOMVS={YES | NO}
BUILDTIMEOUT={ss | 180}
CFGDSN=<configuration file data set name>
COMM={XCF | MQ}
DELAY={ss | 0}
DIAGINFO=dsname
DIAGDUPMSG={nnnnn | 0}
GRPID={xx | '  '}
IOINTERVAL=n
LEOPT={<any>}
LIFECYCLE=500;MY.AGENT.DATA.SET
LOGSTREAM={YES | NO}
MQM=ssid
NUMQTHDS={n | 3}
OVRDELETEDELAY={dd | 0}
PREF=number
PROMPT={YES | NO}
START={COLD | HOT | WARM}
STOPDELAY={ss | 30}
TAKEOVERFILE=name
TAKEOVERTIMEOUT=nn
WLMQUERYINTERVAL={n | 0}
```

**BLOCKOMVS**

This parameter allows you to specify whether the automation manager blocks OMVS shutdown as long as the automation manager is active.

**YES** If BLOCKOMVS=YES is specified, at the automation manager's initialization time, it adds a shutdown block to OMVS. Thus OMVS does not terminate as long as the automation manager is active, even if this is requested by the operator. OMVS is stopped only when the automation manager is stopped with the AM stop command.

Note that a STOP,DEFER causes the automation manager to terminate when all agents connected to it have terminated. Then the stop command for OMVS will get through.

**NO** If BLOCKOMVS=NO is specified and OMVS shuts down , the automation manager abends due to cancellation by OMVS.

**BUILDTIMEOUT**

May be used to specify a time limit for the completion of the data structure build process as used during COLD or WARM start of the primary automation manager. A value from 0-999 seconds may be specified, and a value of 180 (3 minutes) will be assumed if omitted. A specification of 0 suppresses timing of the data structure build process.

**CFGDSN**

The CFGDSN value will be used only on a COLD start, and may be overridden by an initialization prompt response. On other start types, the default CFGDSN will be the one that was in use when automation was last active.

**COMM**

This parameter specifies how communication between the automation manager and the automation agents is realized. The possible values are:

**XCF** Specifies that the automation manager will use XCF for communication with the automation agents. In this case, the takeover file provides the persistent storage medium for holding the current resource states and settings across automation manager sessions.

Using XCF for communication has the following risks:

- All work items travelling to, queued in, or processed by the automation manager are lost when the automation manager terminates abnormally.
- Orders for the automation agents can be broken because some orders could already have been sent at the time when the automation manager terminated abnormally.
- A warmstart is required when an irrecoverable I/O error occurs while reading from or writing to the takeover file.

**MQ** This is the recommended option. It specifies that the automation manager will use WebSphere MQ for communication with the automation agents, and also for holding the status information. With this option, the information in the header of the takeover file determines whether WebSphere MQ or the takeover file is used for a HOT start or takeover.

The COMM parameter and the MQ parameter are mutually dependent. When you specify `COMM=XCF`, the MQ parameter must be left blank. With `COMM=MQ` you must specify an WebSphere MQ subsystem for the MQ parameter.

**DELAY**

Is the number of seconds to be used as a default delay prior to determining the operational mode when the automation manager instance is started. The delay option can be used when you IPL several systems concurrently and want to ensure that the primary or secondary automation manager is started on a particular system.

Note that the DELAY parameter applies only to the IPL of a system, whereas the PREF parameter applies only in the case of a takeover.

A delay value from 0–999 seconds may be specified. A value of 0 (no delay) will be assumed if it is omitted.

This value may by overridden on an individual instance basis by the start command parameter.

This parameter will be ignored when the automation manager instance is started by Automatic Restart Manager or with the specification of TYPE=HOT.

**DIAGDUPMSG**

This is the number of message buffer IDs that are validated before send and after receive. This is for diagnostic purposes. A value for *nnnnn* may be chosen between 0 (no validation) and 99999. The default is 0 and performance decreases with larger values.

**DIAGINFO**

Specifies that the automation manager starts work item recording from the beginning. `dsname` is the name of the data set that will hold the work items. The data set must be a sequential file. It must exist and must be catalogued.

**Note:** The data set name is accepted without checking if the data set exists or if it is accessed by another user.

**GRPID**

Is the same as that currently generated by the SA OS/390 1.3 customization dialog, and will be prefixed with the string INGXSG to create the XCF group name as used by the communication manager function. Note that this value cannot be overridden, and that a null value will be used if not specified. See "Defining the XCF Group" on page 32.

**IOINTERVAL**

This defines the interval that is used to buffer any I/O to the takeover file. The value can be from 0 to 10 seconds. The default is 0 which means that no buffering is done. The maximum is 10 seconds. At the end of the interval any deferred I/O is done.

**LEOPT**

May be used to pass run-time options to the LE environment.

1. Options forced by the Automation Manager

   The following LE runtime options are set by the Automation Manager during initialization:

   ALL31(ON)
   POSIX(ON)

   **Note:** These options must not be overridden by installation default settings (CEEDOPT) with the NONOVR attribute.

2. Default options set by the automation manager during initialization

   The following LE runtime options are set by the automation manager during initialization:

   ```
   ANYHEAP(3M,1M,ANYWHERE,FREE)
   DEPTHCONDLMT(4)
   ERRCOUNT(0)
   HEAP(100M,10M,ANYWHERE,KEEP)
   STACK(64K,64K,ANYWHERE,KEEP)
   STORAGE(NONE,NONE,NONE,128K)
   ```

   **Note:** You may override these options.

3. Recommended LE Options

   The following LE options are recommended for the System Automation Manager:

   ```
      NONIPTSTACK(4K,4K,ANYWHERE,KEEP)
   or THREADSTACK(ON,4K,4K,ANYWHERE,KEEP,512K,128K)
            Note:  NONIPTSTACK was replaced by THREADSTACK in OS/390 LE 2.10
      PROFILE(OFF,'')
      RTLS(OFF)
      STORAGE(NONE,NONE,NONE,128K)
   ```

```
  THREADHEAP(4K,4K,ANYWHERE,KEEP)
  TRACE(OFF,4K,DUMP,LE=0)
  VCTRSAVE(OFF)
  XPLINK(OFF)
```

The following options can be used to gather diagnostic and storage usage information, but should be removed when no longer needed:

```
 RPTSTG(ON)
 RPTOPTS(ON)
```

The LE options below should be tuned using the LE storage reporting facility RPTSTG(ON). The initial value for HEAP storage can be calculated using the following formula:

```
heapsize = 16 MB + (nnn • 8K),
```

where *nnn* is the number of resources and resource groups.

```
ANYHEAP(3M,1M,ANYWHERE,FREE)
HEAP(100M,10M,ANYWHERE,KEEP)
HEAPPOOLS(ON,40,2,64,2,104,2,312,2,624,1,2024,1)
STACK(64K,64K,ANYWHERE,KEEP)
```

The following option is used to direct output created as a result of specifying RPTOPTS(ON) or RPTSTG(ON). It is also used to direct diagnostic messages written to CEEMSG and CEEMOUT by the Automation Manager.

```
MSGFILE(SYSOUT,FBA,121,0,NOENQ)
```

The storage options for the below the line heap need to be tuned.

**Notes:**

1. If an LEOPT= keyword is present in HSAPRM00, it replaces any LEOPT that may have been specified as an input parameter through JCL.
2. When specifying options in HSAPRMxx you may have LEOPT statements on a number of different lines, but the total length of all of the options cannot exceed 4096 characters.

Sample LEOPTS statements are supplied in sample member HSAPRM00.

**LIFECYCLE=***nnnn***;***dataset*

This parameter allows you to prepare for Life Cycle Recording in order to debug automation manager-related problems. Normally, SA z/OS Service will advise when Life Cycle Recording should be enabled. Specify the following:

*nnnn* Defines the size of the data space in number of megabytes (1 through 2097). A value of 500 is recommended and is sufficient in most situations.

*dataset* Specifies the fully-qualified DSN to be used when offloading the dataspace to disk.

**Note:** *nnnn* and *dataset* must be separated by a semicolon without intervening blanks The total length of '*nnnn;dataset*' can be a maximum of 44 bytes.

**LOGSTREAM**

This defines whether or not the automation manager should establish a connection to the system logger at initialization time. The default is YES.

If NO is specified, no access to the log streams HSA.WORKITEM.HISTORY and HSA.MESSAGE.LOG will be established and subsequently no data will be written into them. No work item history besides that shown in the

INGINFO command is available and no detailed information or warning or error messages are available for problem determination.

**MQM** This value specifies the subsystem ID (SSID) of the current WebSphere MQ manager.

The MQ parameter and the COMM parameter are mutually dependent. When you specify `COMM=XCF`, the MQ parameter must be left blank. With `COMM=MQ` you must specify an WebSphere MQ subsystem for the MQ parameter.

**NUMQTHDS**
The NUMQTHDS parameter controls the number of query threads. This value limits the amount of parallel query activity that can be performed. If not specified, a default value of 3 will be used. A maximum of 15 query threads may be specified.

**OVRDELETEDELAY**
Is the number of days that a schedule override should be retained before being automatically deleted. A value of 0 days indicates that schedule overrides are not to be automatically deleted and is the default if no value is specified. A maximum of 366 days may be specified.

**PREF** Specifies the preference given to the instance of the automation manager when determining which of the SAMs should become the primary automation manager.

The value can range from 0 through 15, where 0 is the highest preference. The SAM will only participate in the escalation process when there is no other SAM active with a higher preference. The default is 0.

Note that the PREF parameter applies only in the case of a takeover, whereas the DELAY parameter applies only to the IPL of a system.

**PROMPT**
Specifying YES lets you overwrite the *CFGDSN* parameter (the name of the automation manager configuration file). Message HSAM1302A will come up and wait for a response. You may now specify the keyword/value pair

```
CFGDSN=<fully.qualified.data.set.name>
```

or you may use a null or ’U’ response to indicate no override values are to be applied.

**START**
Defines the start mode of the automation manager. During initialization, the automation manager retrieves input from:

**1** Parameter CFGDSN

**2** Schedule overrides

**3** Persistent data store (votes, triggers, resource states)

The following table shows where the automation manager retrieves initialization data for the possible values for parameter START.

| | COLD | WARM | HOT |
|---|---|---|---|
| 1 | name of automation manager configuration file is taken from PARMLIB, the START command or via PROMPT=YES option | last used value taken | last used value taken |
| 2 | deleted | taken from last run | taken from last run |
| 3 | deleted | deleted | taken from last run |

***Recommendation:*** Use COLD for the very first time, or when the schedule override file should be cleared. Use WARM if the automation policy has changed, that is, the automation manager configuration file has been rebuilt. Use HOT in any other case.

The start mode does not affect the secondary automation managers.

The START parameter can also be specified in the Automation Manager JCL. If the HSAPRM00 values are to be used, the START= parameter must be removed from the JCL.

**STOPDELAY**

Is the number of seconds to be used when an MVS F <jobname>,STOP,DEFER command is entered for the primary automation manager. This delay will be invoked only if one or more secondary automation managers are active and ready when the command is received.

**TAKEOVERFILE**

This defines the data set name of the takeover file. It must be fully qualified.

Note that if the HSPRM*xx* member is shared among systems in the same XCF group and these systems host downlevel automation managers (lower than SA z/OS 2.3), APAR OA02723 must be installed on the downlevel systems. Otherwise the TAKEOVERFILE parameter will be rejected by means of message HSAM5200E and the automation manager is terminated.

**TAKEOVERTIMEOUT**

*nn* may range from 1 to 600 seconds. The default is 12 seconds.

If communication is MQ:

If the (secondary) automation manager performs a takeover, or an automation manager is started HOT, it is examined whether WebSphere MQ is ready. If this is not the case, the automation manager enters a retry loop. The TAKEOVERTIMEOUT parameter determines how many seconds the automation manager should wait (retry) until it switches from mode=HOT to mode=WARM.

If communication is XCF:

If the (secondary) automation manager performs a takeover, or an automation manager is started HOT, it will wait for specified seconds before the takeover is done from the takeover file. This delay may be required in order to allow VSAM to perform its cleanup activities on the takeover file.

**WLMQUERYINTERVAL**

This specifies the time in minutes between queries of WLM by the automation manager, as used for resource aware application move. The default is 0, which means that no querying of WLM is done. The valid range for WLMQUERYINTERVAL is from 0 to 600 minutes (that is, 10 hours).

## Syntax for HSAPRM00

# Appendix H. INGDLG Command

The INGDLG command allocates required DD names and invokes the customization dialog. Its syntax is:

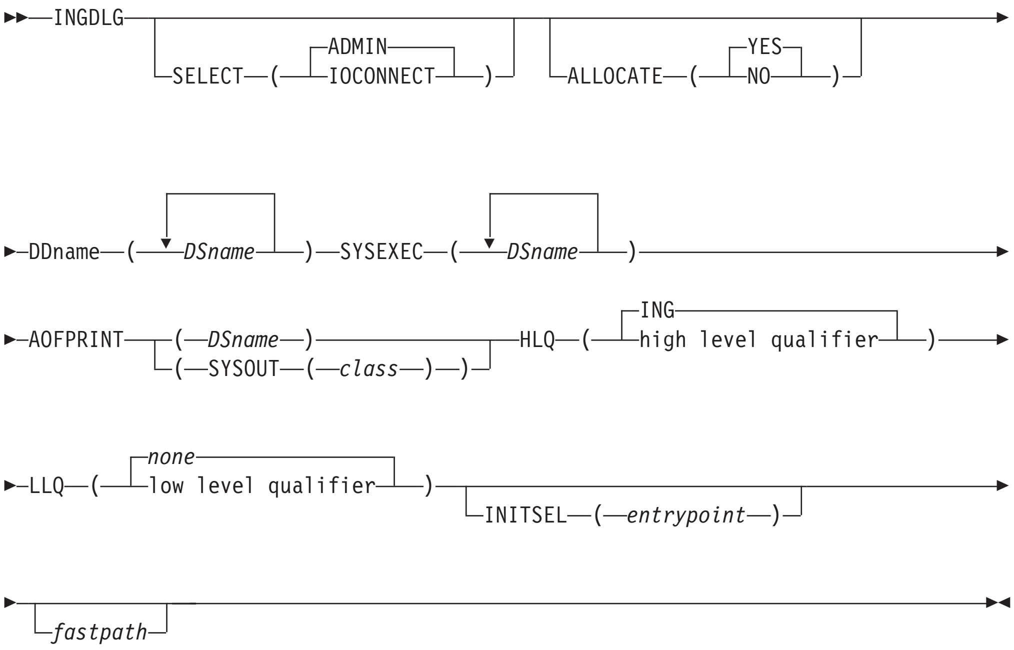


*Figure 30. INGDLG Command Syntax*

Its parameters are:

**SELECT**
Enables you to select either ADMIN or IOCONNECT. If the SELECT keyword is not specified, SELECT (ADMIN) is the default.

**ADMIN**
Enables the selection of automation policy dialogs. This is the default.

**IOCONNECT**
Enables the selection of I/O connectivity dialogs

**ALLOCATE**
Controls defining DD names. If ALLOCATE is not specified, ALLOCATE (YES) is the default.

**YES**
Allocates the necessary libraries according to the specifications in HLQ and LLQ parameters.

If DDname AOFTABL is specified as an additional parameter, that data set is also allocated for ISPTLIB.

Furthermore, to avoid enqueue situations for multiple users, the name of the ISPF profile data set is obtained and allocated as the first data set of the table input library.

**NO** Does not perform any allocation of data sets. The libraries needed for the customization dialog need to be allocated prior to invocation of INGDLG.

**DDname (DSname)**

Fully qualified data set name to associate. Prefixes and suffixes defined using this panel are not appended to this name. For the DD name `AOFPRINT`, the following syntax is also valid:

```
AOFPRINT(SYSOUT(class))
```

, where *class* is a valid output class, creating a DD statement with SYSOUT=class. In this case, the output is placed into the JES output class *class*.

DD name AOFTABL is required if policies are changed (option 0.10) or data sets for batch functions are specified (option 0.11). This data is written to the data set that is allocated to AOFTABL.

**SYSEXEC(DSname DSname DSname ...)**

For DD name `SYSEXEC`, multiple data set names are supported: `SYSEXEC(DSname DSname DSname ...)`. This will result in the following command:

```
TSO ALLOC ALTLIB ACTIVATE APPLICATION(EXEC)
        DATASET(DSname DSname DSname ...) UNCOND
```

**AOFPRINT**

For DD name `AOFPRINT`, *DSname* is a fully qualified data set name or a request to allocate a SYSOUT data set: AOFPRINT(SYSOUT(class))

**HLQ** Enables you to change the high level qualifier (HLQ) of the SMP/E data sets, which currently is `ING`, to a HLQ of your choice. If you do not specify this parameter, `ING` is retained as the default.

**LLQ** Enables you to establish a suffix for default data set names (The default is none).

**INITSEL**

may be used to provide a user-selected entry point to the customization dialog. If this keyword is specified, you will not see the Customization Dialog Primary Menu as the first panel when invoking the customization dialog, but it provides a fast path to some other panel, for example, the Entry Name Selection panel for a frequently used entry type. Valid values are those that you can specify as fast path in the customization dialog, for example, to reach the APPC application:

```
=APL; S APPC
```

or to reach application group ***CICS_APG***:

```
=APG; S CICS_APG
```

or to just reach the Entry Name Selection panel for Applications:

```
=APL;
```

**fastpath**

Any words that are not the reserved keywords. The fastpath words are passed as parameters to I/O operations dialogs, if selected.

Return codes for this routine are:

**0** No errors encountered

**4** ISPF is not active

**8** Error in data set allocation

**12** Error in data set de-allocation or a failed allocation

# Glossary

This glossary includes terms and definitions from:

- The *IBM Dictionary of Computing* New York: McGraw-Hill, 1994.
- The *American National Standard Dictionary for Information Systems* , ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies can be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions are identified by the symbol (A) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

The following cross-references are used in this glossary:

**Contrast with.** This refers to a term that has an opposed or substantively different meaning.
**Deprecated term for.** This indicates that the term should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.
**See.** This refers the reader to multiple-word terms in which this term appears.
**See also.** This refers the reader to terms that have a related, but not synonymous, meaning.
**Synonym for.** This indicates that the term has the same meaning as a preferred term, which is defined in the glossary.
**Synonymous with.** This is a backward reference from a defined term to all other terms that have the same meaning.

## A

**ACF.** Automation control file.

**ACF/NCP.** Advanced Communications Function for the Network Control Program. See *Advanced Communications Function* and *Network Control Program*.

**ACF/VTAM.** Advanced Communications Function for the Virtual Telecommunications Access Method. Synonym for *VTAM*. See *Advanced Communications Function* and *Virtual Telecommunications Access Method*.

**active monitoring.** In SA z/OS, the acquiring of resource status information by soliciting such information at regular, user-defined intervals. See also *passive monitoring*.

**adapter.** Hardware card that enables a device, such as a workstation, to communicate with another device, such as a monitor, a printer, or some other I/O device.

**Address Space Workflow.** In RMF, a measure of how a job uses system resources and the speed at which the job moves through the system. A low workflow indicates that a job has few of the resources it needs and is contending with other jobs for system resources. A high workflow indicates that a job has all the resources it needs to execute.

**adjacent hosts.** Systems connected in a peer relationship using adjacent NetView sessions for purposes of monitoring and control.

**adjacent NetView.** In SA z/OS, the system defined as the communication path between two SA z/OS systems that do not have a direct link. An adjacent NetView is used for message forwarding and as a communication link between two SA z/OS systems. For example, the adjacent NetView is used when sending responses from a focal point to a remote system.

**Advanced Communications Function (ACF).** A group of IBM licensed programs (principally VTAM, TCAM, NCP, and SSP) that use the concepts of Systems Network Architecture (SNA), including distribution of function and resource sharing.

**advanced program-to-program communication (APPC).** A set of inter-program communication services that support cooperative transaction processing in a Systems Network Architecture (SNA) network. APPC is the implementation, on a given system, of SNA's logical unit type 6.2.

**alert.** (1) In SNA, a record sent to a system problem management focal point or to a collection point to communicate the existence of an alert condition. (2) In NetView, a high-priority event that warrants immediate attention. A database record is generated for certain event types that are defined by user-constructed filters.

**alert condition.** A problem or impending problem for which some or all of the process of problem determination, diagnosis, and resolution is expected to require action at a control point.

**alert focal-point system.** See entry for NPDA focal-point system under *focal-point system*.

**alert threshold.** An application or volume service value that determines the level at which SA z/OS changes the associated icon in the graphical interface to the alert color. SA z/OS may also issue an alert. See *warning threshold*.

**AMC.** (1) Automation Manager Configuration (2) The Auto Msg Classes entry type

**APF.** Authorized program facility.

**API.** Application programming interface.

**APPC.** Advanced program-to-program communications.

**application.** An z/OS subsystem or job monitored by SA z/OS.

**Application entry.** A construct, created with the customization dialogs, used to represent and contain policy for an application.

**application group.** A named set of applications. An application group is part of an SA z/OS enterprise definition and is used for monitoring purposes.

**ApplicationGroup entry.** A construct, created with the customization dialogs, used to represent and contain policy for an application group.

**application program.** (1) A program written for or by a user that applies to the user's work, such as a program that does inventory or payroll. (2) A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

**ARM.** Automatic restart management.

**ASCB.** Address space control block.

**ASCB status.** An application status derived by SA z/OS running a routine (the ASCB checker) that searches the z/OS address space control blocks (ASCBs) for address spaces with a particular job name. The job name used by the ASCB checker is the job name defined in the customization dialog for the application.

**ASCII (American National Standard Code for Information Interchange).** The standard code, using a coded character set consisting of 7-bit coded characters (8-bit including parity check), for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (A)

**ASF.** Automation status file.

**authorized program facility (APF).** A facility that permits identification of programs that are authorized to use restricted functions.

**automated function.** SA z/OS automated functions are automation operators, NetView autotasks that are assigned to perform specific automation functions. However, SA z/OS defines its own synonyms, or *automated function names*, for the NetView autotasks, and these function names are referred to in the sample policy databases provided by SA z/OS. For example, the automation operator AUTBASE corresponds to the SA z/OS automated function BASEOPER.

**automated console operations (ACO).** The concept (versus a product) of using computers to perform a large subset of tasks ordinarily performed by operators, or assisting operators in performing these tasks.

**automatic restart management (ARM).** A z/OS recovery function that improves the availability of specified subsystems and applications by automatically restarting them under certain circumstances. Automatic restart management is a function of the Cross-System Coupling Facility (XCF) component of z/OS.

**automatic restart management element name.** In MVS 5.2 or later, z/OS automatic restart management requires the specification of a unique sixteen character name for each address space that registers with it. All automatic restart management policy is defined in terms of the element name, including SA z/OS's interface with it.

**automation.** The automatic initiation of actions in response to detected conditions or events. SA z/OS provides automation for z/OS applications, z/OS components, and remote systems that run z/OS. SA z/OS also provides tools that can be used to develop additional automation.

**automation agent.** In SA z/OS, the automation function is split up between the automation manager and the automation agents. The observing, reacting and doing parts are located within the NetView address space, and are known as the ***automation agents***. The automation agents are responsible for:

- recovery processing
- message processing
- active monitoring: they propagate status changes to the automation manager

**automation configuration file.** The data set that consists of:

- the automation control file (ACF)
- the automation manager configuration file (AMC)
- the NetView automation table (AT)
- the MPFLSTSA member

**automation control file (ACF).** In SA z/OS, a file that contains system-level automation policy information. There is one master automation control file for each NetView system on which SA z/OS is installed. Additional policy information and all resource status information is contained in the policy database (PDB). The SA z/OS customization dialogs must be used to build the automation control files. They must not be edited manually.

**automation flags.** In SA z/OS, the automation policy settings that determine the operator functions that are automated for a resource and the times during which automation is active. When SA z/OS is running, automation is controlled by automation flag policy settings and override settings (if any) entered by the operator. Automation flags are set using the customization dialogs.

**automation manager.** In SA z/OS, the automation function is split up between the automation manager and the automation agents. The coordination, decision making and controlling functions are processed by each sysplex's ***automation manager***.

The automation manager contains a model of all of the automated resources within the sysplex. The automation agents feed the automation manager with status information and perform the actions that the automation manager tells them to.

The automation manager provides ***sysplex-wide*** automation.

**Automation Manager Configuration.** The Automation Manager Configuration file (AMC) contains an image of the automated systems in a sysplex or of a standalone system.

**Automation NetView.** In SA z/OS the NetView that performs routine operator tasks with command procedures or uses other ways of automating system and network management, issuing automatic responses to messages and management services units.

**automation operator.** NetView automation operators are NetView autotasks that are assigned to perform specific automation functions. See also *automated function*. NetView automation operators may receive messages and process automation procedures. There are no logged-on users associated with automation operators. Each automation operator is an operating system task and runs concurrently with other NetView tasks. An automation operator could be set up to handle JES2 messages that schedule automation procedures, and an automation statement could route such messages to the automation operator. Similar to *operator station task*. SA z/OS message monitor tasks and target control tasks are automation operators.

**automation policy.** The policy information governing automation for individual systems. This includes automation for applications, z/OS subsystems, z/OS data sets, and z/OS components.

**automation policy settings.** The automation policy information contained in the automation control file. This information is entered using the customization dialogs. You can display or modify these settings using the customization dialogs.

**automation procedure.** A sequence of commands, packaged as a NetView command list or a command processor written in a high-level language. An automation procedure performs automation functions and runs under NetView.

**automation status file.** In SA z/OS, a file containing status information for each automated subsystem, component or data set. This information is used by SA z/OS automation when taking action or when determining what action to take. In Release 2 and above of AOC/MVS, status information is also maintained in the operational information base.

**automation table (AT).** See *NetView automation table*.

**autotask.** A NetView automation task that receives messages and processes automation procedures. There are no logged-on users associated with autotasks. Each autotask is an operating system task and runs concurrently with other NetView tasks. An autotask could be set up to handle JES2 messages that schedule automation procedures, and an automation statement could route such messages to the autotasks. Similar to *operator station task*. SA z/OS message monitor tasks and target control tasks are autotasks. Also called *automation operator*.

**available.** In VTAM programs, pertaining to a logical unit that is active, connected, enabled, and not at its session limit.

# B

**basic mode.** A central processor mode that does not use logical partitioning. Contrast with *logically partitioned (LPAR) mode*.

**BCP Internal Interface.** Processor function of CMOS-390, zSeries processor families. It allows the communication between basic control programs such as z/OS and the processor support element in order to exchange information or to perform processor control functions. Programs using this function can perform hardware operations such as ACTIVATE or SYSTEM RESET.

**beaconing.** The repeated transmission of a frame or messages (beacon) by a console or workstation upon detection of a line break or outage.

**BookManager.** An IBM product that lets users view softcopy documents on their workstations.

# C

**central processor (CP).** The part of the computer that contains the sequencing and processing facilities for instruction execution, initial program load (IPL), and other machine operations.

**central processor complex (CPC).** A physical collection of hardware that consists of central storage, one or more central processors, timers, and channels.

**central site.** In a distributed data processing network, the central site is usually defined as the focal point for alerts, application design, and remote system management tasks such as problem management.

**CFR/CFS and ISC/ISR.** I/O operations can display and return data about integrated system channels (ISC) connected to a coupling facility and coupling facility receiver (CFR) channels and coupling facility sender (CFS) channels.

**channel.** A path along which signals can be sent; for example, data channel, output channel. See also *link*.

**channel path identifier.** A system-unique value assigned to each channel path.

**CHPID.** In SA z/OS, channel path ID; the address of a channel.

**CHPID port.** A label that describes the system name, logical partitions, and channel paths.

**channel-attached.** (1) Attached directly by I/O channels to a host processor (for example, a channel-attached device). (2) Attached to a controlling unit by cables, rather than by telecommunication lines. Contrast with *link-attached*. Synonymous with *local*.

**CI.** Console integration.

**CICS/VS.** Customer Information Control System for Virtual Storage.

**CLIST.** Command list.

**clone.** A set of definitions for application instances that are derived from a basic application definition by substituting a number of different system-specific values into the basic definition.

**clone ID.** A generic means of handling system-specific values such as the MVS SYSCLONE or the VTAM subarea number. Clone IDs can be substituted into application definitions and commands to customize a basic application definition for the system that it is to be instantiated on.

**CNC.** A channel path that transfers data between a host system image and an ESCON control unit. It can be point-to-point or switchable.

**command.** A request for the performance of an operation or the execution of a particular program.

**command facility.** The component of NetView that is a base for command processors that can monitor, control, automate, and improve the operation of a network. The successor to NCCF.

**command list (CLIST).** (1) A list of commands and statements, written in the NetView command list language or the REXX language, designed to perform a specific function for the user. In its simplest form, a command list is a list of commands. More complex command lists incorporate variable substitution and conditional logic, making the command list more like a conventional program. Command lists are typically interpreted rather than being compiled. (2) In SA z/OS, REXX command lists that can be used for automation procedures.

**command procedure.** In NetView, either a command list or a command processor.

**command processor.** A module designed to perform a specific function. Command processors, which can be written in assembler or a high-level language (HLL), are issued as commands.

**Command Tree/2.** An OS/2-based program that helps you build commands on an OS/2 window, then routes the commands to the destination you specify (such as a 3270 session, a file, a command line, or an application program). It provides the capability for operators to build commands and route them to a specified destination.

**common commands.** The SA z/OS subset of the CPC operations management commands.

**common routine.** One of several SA z/OS programs that perform frequently used automation functions. Common routines can be used to create new automation procedures.

**Common User Access (CUA) architecture.** Guidelines for the dialog between a human and a workstation or terminal.

**communication controller.** A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit or by a program executed in a processor to which the controller is connected. It manages the details of line control and the routing of data through a network.

**communication line.** Deprecated term for *telecommunication line*.

**connectivity view.** In SA z/OS, a display that uses graphic images for I/O devices and lines to show how they are connected.

**console automation.** The process of having NetView facilities provide the console input usually handled by the operator.

**console connection.** In SA z/OS, the 3270 or ASCII (serial) connection between a PS/2 computer and a target system. Through this connection, the workstation appears (to the target system) to be a console.

**console integration (CI).** A hardware facility that if supported by an operating system, allows operating system messages to be transferred through an internal hardware interface for display on a system console. Conversely, it allows operating system commands entered at a system console to be transferred through an internal hardware interface to the operating system for processing.

**consoles.** Workstations and 3270-type devices that manage your enterprise.

**Control units.** Hardware units that control I/O operations for one or more devices. You can view information about control units through I/O operations, and can start or stop data going to them by blocking and unblocking ports.

**controller.** A unit that controls I/O operations for one or more devices.

**couple data set.** A data set that is created through the XCF couple data set format utility and, depending on its designated type, is shared by some or all of the z/OS systems in a sysplex. See also *sysplex couple data set* and *XCF couple data set*.

**coupling facility.** The hardware element that provides high-speed caching, list processing, and locking functions in a sysplex.

**CP.** Central processor.

**CPC.** Central processor complex.

**CPC operations management commands.** A set of commands and responses for controlling the operation of System/390 CPCs.

**CPC subset.** All or part of a CPC. It contains the minimum *resource* to support a single control program.

**CPCB.** Command processor control block; an I/O operations internal control block that contains information about the command being processed.

**CPU.** Central processing unit. Deprecated term for *processor*.

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

**CTC.** The channel-to-channel (CTC) channel can communicate with a CTC on another host for intersystem communication.

**Customer Information Control System (CICS).** A general-purpose transactional program that controls online communication between terminal users and a database for a large number of end users on a real-time basis.

**customization dialogs.** The customization dialogs are an ISPF application. They are used to customize the enterprise policy, like, for example, the enterprise resources and the relationships between resources, or the automation policy for systems in the enterprise. How to use these dialogs is described in *IBM Tivoli System Automation for z/OS Customizing and Programming*.

**CVC.** A channel operating in converted (CVC) mode transfers data in blocks and a CBY channel path transfers data in bytes. Converted CVC or CBY channel paths can communicate with a parallel control unit. This resembles a point-to-point parallel path and dedicated connection, regardless whether it passes through a switch.

# D

**DASD.** Direct access storage device.

**data services task (DST).** The NetView subtask that gathers, records, and manages data in a VSAM file or a network device that contains network management information.

**data set.** The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

**data set members.** Members of partitioned data sets that are individually named elements of a larger file that can be retrieved by name.

**DBCS.** Double-byte character set.

**DCCF.** Disabled console communication facility.

**DCF.** Document composition facility.

**DELAY Report.** An RMF report that shows the activity of each job in the system and the hardware and software resources that are delaying each job.

**Devices.** You can see information about all devices (such as printers, tape or disk drives, displays, or communications controllers) attached to a particular switch, and control paths and jobs to devices.

**DEVR Report.** An RMF report that presents information about the activity of I/O devices that are delaying jobs.

**dialog.** Interactive 3270 panels.

**direct access storage device (DASD).** A device in which the access time is effectively independent of the location of the data; for example, a disk.

**disabled console communication facility (DCCF).** A z/OS component that provides limited-function console communication during system recovery situations.

**display.** (1) To present information for viewing, usually on the screen of a workstation or on a hardcopy device. (2) Deprecated term for *panel*.

**disk operating system (DOS).** (1) An operating system for computer systems that use disks and diskettes for auxiliary storage of programs and data. (2) Software for a personal computer that controls the processing of programs. For the IBM Personal Computer, the full name is Personal Computer Disk Operating System (PCDOS).

**distribution manager.** The component of the NetView program that enables the host system to use, send, and delete files and programs in a network of computers.

**domain.** (1) An access method and its application programs, communication controllers, connecting lines, modems, and attached workstations. (2) In SNA, a system services control point (SSCP) and the physical units (PUs), logical units (LUs), links, link stations, and associated resources that the SSCP can control by means of activation requests and deactivation requests.

**double-byte character set (DBCS).** A character set, such as Kanji, in which each character is represented by a 2-byte code.

**DP enterprise.** Data processing enterprise.

**DSIPARM.** This file is a collection of members of NetView's customization.

**DST.** Data Services Task.

# E

**EBCDIC.** Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

**ECB.** Event control block. A control block used to represent the status of an event.

**EMCS.** Extended multiple console support.

**enterprise.** An organization, such as a business or a school, that uses data processing.

**enterprise monitoring.** Enterprise monitoring is used by SA z/OS to update the *NetView Management Console (NMC)* resource status information that is stored in the *Resource Object Data Manager (RODM)*. Resource status information is acquired by enterprise monitoring of the *Resource Measurement Facility (RMF) Monitor III* service information at user-defined intervals. SA z/OS stores this information in its operational information base, where it is used to update the information presented to the operator in graphic displays.

**entries.** Resources, such as processors, entered on panels.

**entry type.** Resources, such as processors or applications, used for automation and monitoring.

**environment.** Data processing enterprise.

**error threshold.** An automation policy setting that specifies when SA z/OS should stop trying to restart or recover an application, subsystem or component, or offload a data set.

**ESA.** Enterprise Systems Architecture.

**eServer.** Processor family group designator used by the SA z/OS customization dialogs to define a target hardware as member of the zSeries or 390-CMOS processor families.

**event.** (1) In NetView, a record indicating irregularities of operation in physical elements of a network. (2) An occurrence of significance to a task; for example, the completion of an asynchronous operation, such as an input/output operation. (3) Events are part of a trigger condition, in a way that if all events of a trigger condition have occurred, a *STARTUP* or *SHUTDOWN* of an application is performed.

**exception condition.** An occurrence on a system that is a deviation from normal operation. SA z/OS monitoring highlights exception conditions and allows an SA z/OS enterprise to be managed by exception.

**extended recovery facility (XRF).** A facility that minimizes the effect of failures in z/OS, VTAM, the host processor, or high availability applications during sessions between high availability applications and designated terminals. This facility provides an alternate subsystem to take over sessions from the failing subsystem.

# F

**fallback system.** See *secondary system*.

**field.** A collection of bytes within a record that are logically related and are processed as a unit.

**file manager commands.** A set of SA z/OS commands that read data from or write data to the automation control file or the operational information base. These commands are useful in the development of automation that uses SA z/OS facilities.

**focal point.** In NetView, the focal-point domain is the central host domain. It is the central control point for any management services element containing control of the network management data.

**focus host.** A processor with the role in the context of a unified system image

**focal point system.** (1) A system that can administer, manage, or control one or more target systems. There are a number of different focal point system associated with IBM automation products. (2) **NMC focal point system**. The NMC focal point system is a NetView system with an attached workstation server and LAN that gathers information about the state of the network. This focal point system uses RODM to store the data it collects in the data model. The information stored in RODM can be accessed from any LAN-connected workstation with NetView Management Console installed. (3) **NPDA focal point system.** This is a NetView system that collects all the NPDA alerts that are generated within your enterprise. It is supported by NetView. If you have SA z/OS installed the NPDA focal point system must be the same as your NMC focal point system. The NPDA focal point system is also known as the *alert focal point system*. (4) **SA z/OS Processor Operations focal point system.** This is a NetView system that has SA z/OS host code installed. The SA z/OS Processor Operations focal point system receives messages from the systems and operator consoles of the machines that it controls. It provides full systems and operations console function for its target systems. It can be used to IPL these systems. Note that some restrictions apply to the Hardware Management Console for an S/390 microprocessor cluster. (5) **SA z/OS SDF focal point system.** The SA z/OS SDF focal point system is an SA z/OS NetView system that collects status information from other SA z/OS NetViews within your enterprise. (6) **Status focal point system.** In NetView, the system to which STATMON, VTAM and NLDM send status information on network resources. If you have a NMC focal point, it must be on the same system as the Status focal point. (7) **Hardware Management Console.** Although not listed as a focal point, the Hardware Management Console acts as a focal point for the console functions of an S/390 microprocessor cluster. Unlike all the other focal points in this definition, the Hardware Management Console runs on a LAN-connected workstation,

**frame.** For a System/390 microprocessor cluster, a frame contains one or two central processor complexes (CPCs), support elements, and AC power distribution.

**full-screen mode.** In NetView, a form of panel presentation that makes it possible to display the contents of an entire workstation screen at once. Full-screen mode can be used for fill-in-the-blanks prompting. Contrast with *line mode*.

# G

**gateway session.** An NetView-NetView Task session with another system in which the SA z/OS outbound gateway operator logs onto the other NetView session without human operator intervention. Each end of a gateway session has both an inbound and outbound gateway operator.

**generic alert.** Encoded alert information that uses code points (defined by IBM and possibly customized by users or application programs) stored at an alert receiver, such as NetView.

**generic routines.** In SA z/OS, a set of self-contained automation routines that can be called from the NetView automation table, or from user-written automation procedures.

**group.** A collection of target systems defined through configuration dialogs. An installation might set up a group to refer to a physical site or an organizational or application entity.

**group entry.** A construct, created with the customization dialogs, used to represent and contain policy for a group.

**group entry type.** A collection of target systems defined through the customization dialog. An installation might set up a group to refer to a physical site or an organizational entity. Groups can, for example, be of type STANDARD or SYSPLEX.

# H

**Hardware Management Console.** A console used by the operator to monitor and control a System/390 microprocessor cluster.

**Hardware Management Console Application (HWMCA).** A direct-manipulation object-oriented graphical user interface that provides single point of control and single system image for hardware elements. HWMCA provides customer grouping support, aggregated and real-time system status using colors, consolidated hardware messages support, consolidated operating system messages support, consolidated service support, and hardware commands targeted at a single system, multiple systems, or a customer group of systems.

**heartbeat.** In SA z/OS, a function that monitors the validity of the status forwarding path between remote systems and the NMC focal point, and monitors the availability of remote z/OS systems, to ensure that status information displayed on the SA z/OS workstation is current.

**help panel.** An online panel that tells you how to use a command or another aspect of a product.

**hierarchy.** In the NetView program, the resource types, display types, and data types that make up the organization, or levels, in a network.

**high-level language (HLL).** A programming language that does not reflect the structure of any particular computer or operating system. For the NetView program, the high-level languages are PL/I and C.

**HLL.** High-level language.

**host system.** In a coupled system or distributed system environment, the system on which the facilities for centralized automation run. SA z/OS publications refer to target systems or focal-point systems instead of hosts.

**host (primary processor).** The processor at which you enter a command (also known as the *issuing processor*).

**HWMCA.** Hardware Management Console Application. Application for the graphic hardware management console that monitors and controls a central processor complex. It is attached to a target processor (a system 390 microprocessor cluster) as a dedicated system console. This microprocessor uses OCF to process commands.

## I

**images.** A grouping of processors and I/O devices that you define. You can define a single-image mode that allows a multiprocessor system to function as one central processor image.

**IMS/VS.** Information Management System/Virtual Storage.

**inbound.** In SA z/OS, messages sent to the focal-point system from the PC or target system.

**inbound gateway operator.** The automation operator that receives incoming messages, commands, and responses from the outbound gateway operator at the sending system. The inbound gateway operator handles communications with other systems using a gateway session.

**Information Management System/Virtual Storage (IMS/VS).** A database/data communication (DB/DC) system that can manage complex databases and networks. Synonymous with IMS.

**INGEIO PROC.** The I/O operations default procedure name; part of the SYS1.PROCLIB.

**initial program load (IPL).** (1) The initialization procedure that causes an operating system to commence operation. (2) The process by which a configuration image is loaded into storage at the beginning of a workday or after a system malfunction. (3) The process of loading system programs and preparing a system to run jobs.

**initialize automation.** SA z/OS-provided automation that issues the correct z/OS start command for each subsystem when SA z/OS is initialized. The automation ensures that subsystems are started in the order specified in the automation control file and that prerequisite applications are functional.

**input/output support processor (IOSP).** The hardware unit that provides I/O support functions for the primary support processor and maintenance support functions for the processor controller.

**Interactive System Productivity Facility (ISPF).** An IBM licensed program that serves as a full-screen editor and dialog manager. Used for writing application programs, it provides a means of generating standard screen panels and interactive dialogs between the application programmer and the terminal user.

**interested operator list.** The list of operators who are to receive messages from a specific target system.

**internal token.** A *logical token* (LTOK); name by which the I/O resource or object is known; stored in IODF.

**IOCDS.** I/O configuration data set. The data set that describes the I/O configuration.

**I/O Ops.** I/O operations.

**IOSP.** Input/Output Support Processor.

**I/O operations.** The part of SA z/OS that provides you with a single point of logical control for managing connectivity in your active I/O configurations. I/O operations takes an active role in detecting unusual conditions and lets you view and change paths between a processor and an I/O device, using dynamic switching (the ESCON director). Also known as I/O Ops.

**I/O resource number.** Combination of channel path identifier (CHPID), device number, etc. See internal token.

**IPL.** Initial program load.

**ISA.** Industry Standard Architecture.

**ISPF.** Interactive System Productivity Facility.

**ISPF console.** From this 3270-type console you are logged onto ISPF to use the runtime panels for I/O operations and SA z/OS customization panels.

**issuing host.** See *primary host*; the base program at which you enter a command for processing.

## J

**JCL.** Job control language.

**JES.** Job entry subsystem.

**job.** (1) A set of data that completely defines a unit of work for a computer. A job usually includes all necessary computer programs, linkages, files, and instructions to the operating system. (2) An address space.

**job control language (JCL).** A problem-oriented language designed to express statements in a job that are used to identify the job or describe its requirements to an operating system.

**job entry subsystem (JES).** A facility for spooling, job queuing, and managing I/O. In SA z/OS publications, JES refers to JES2 or JES3, unless distinguished as being either one or the other.

## K

**Kanji.** An ideographic character set used in Japanese. See also *double-byte character set*.

## L

**LAN.** Local area network.

**line mode.** A form of screen presentation in which the information is presented a line at a time in the message area of the terminal screen. Contrast with *full-screen mode*.

**link.** (1) In SNA, the combination of the link connection and the link stations joining network nodes; for example, a System/370 channel and its associated protocols, a serial-by-bit connection under the control of synchronous data link control (SDLC). (2) In SA z/OS, link connection is the physical medium of transmission.

**link-attached.** Describes devices that are physically connected by a telecommunication line. Contrast with *channel-attached*.

**Linux for zSeries and S/390.** UNIX-like open source operating system conceived by Linus Torvalds and developed across the internet.

**local.** Pertaining to a device accessed directly without use of a telecommunication line. Synonymous with *channel-attached*.

**local area network (LAN).** (1) A network in which a set of devices is connected for communication. They can be connected to a larger network. See also *token ring*. (2) A network in which communications are limited to a moderately-sized geographic area such as a single office building, warehouse, or campus, and that do not generally extend across public rights-of-way.

**logical partition (LP).** A subset of the processor hardware that is defined to support an operating system. See also *logically partitioned (LPAR) mode*.

**logical switch number (LSN).** Assigned with the switch parameter of the CHPID macro of the IOCP.

**logical token (LTOK).** Resource number of an object in the IODF.

**logical unit (LU).** In SNA, a port through which an end user accesses the SNA network and the functions provided by system services control points (SSCPs). An LU can support at least two sessions — one with an SSCP and one with another LU — and may be capable of supporting many sessions with other LUs. See also *physical unit (PU)* and *system services control point (SSCP)*.

**logical unit (LU) 6.2.** A type of logical unit that supports general communications between programs in a distributed processing environment. LU 6.2 is characterized by (a) a peer relationship between session partners, (b) efficient use of a session for multiple transactions, (c) comprehensive end-to-end error processing, and (d) a generic application program interface (API) consisting of structured verbs that are mapped into a product implementation. Synonym for advanced program-to-program communications (APPC).

**logically partitioned (LPAR) mode.** A central processor mode that enables an operator to allocate system processor hardware resources among several logical partitions. Contrast with *basic mode*.

**LOGR.** The sysplex logger.

**LP.** Logical partition.

**LPAR.** Logically partitioned (mode).

**LU.** Logical unit.

**LU-LU session.** In SNA, a session between two logical units (LUs) in an SNA network. It provides communication between two end users, or between an end user and an LU services component.

**LU 6.2.** Logical unit 6.2.

**LU 6.2 session.** A session initiated by VTAM on behalf of an LU 6.2 application program, or a session initiated by a remote LU in which the application program specifies that VTAM is to control the session by using the APPCCMD macro.

## M

**MAT.** Deprecated term for NetView Automation Table.

**MCA.** Micro Channel* architecture.

**MCS.** Multiple console support.

**member.** A specific function (one or more modules/routines) of a multisystem application that is defined to XCF and assigned to a group by the multisystem application. A member resides on one system in the sysplex and can use XCF services to communicate (send and receive data) with other members of the same group.

**message automation table (MAT).** Deprecated term for NetView Automation Table.

**message class.** A number that SA z/OS associates with a message to control routing of the message. During automated operations, the classes associated with each message issued by SA z/OS are compared to the classes assigned to each notification operator. Any operator with a class matching one of the message's classes receives the message.

**message forwarding.** The SA z/OS process of sending messages generated at an SA z/OS target system to the SA z/OS focal-point system.

**message group.** Several messages that are displayed together as a unit.

**message monitor task.** A task that starts and is associated with a number of communications tasks. Message monitor tasks receive inbound messages from a communications task, determine the originating target system, and route the messages to the appropriate target control tasks.

**message processing facility (MPF).** A z/OS table that screens all messages sent to the z/OS console. The MPF compares these messages with a customer-defined list of messages on which to automate, suppress from the z/OS console display, or both, and marks messages to automate or suppress. Messages are then broadcast on the subsystem interface (SSI).

**message suppression.** The ability to restrict the amount of message traffic displayed on the z/OS console.

**Micro Channel architecture.** The rules that define how subsystems and adapters use the Micro Channel bus in a computer. The architecture defines the services that each subsystem can or must provide.

**microprocessor.** A processor implemented on one or a small number of chips.

**migration.** Installation of a new version or release of a program to replace an earlier version or release.

**MP.** Multiprocessor.

**MPF.** Message processing facility.

**MPFLSTSA.** The MPFLST member that is built by SA z/OS.

**Multiple Virtual Storage (MVS).** An IBM licensed program. MVS, which is the predecessor of OS/390, is an operating system that controls the running of programs on a System/390 or System/370 processor. MVS includes an appropriate level of the Data Facility Product (DFP) and Multiple Virtual Storage/Enterprise Systems Architecture System Product Version 5 (MVS/ESA SP5).

**multiprocessor (MP).** A CPC that can be physically partitioned to form two operating processor complexes.

**multisystem application.** An application program that has various functions distributed across z/OS images in a multisystem environment.

**multisystem environment.** An environment in which two or more z/OS images reside in one or more processors, and programs on one image can communication with programs on the other images.

**MVS.** Multiple Virtual Storage, predecessor of z/OS.

**MVS image.** A single occurrence of the MVS/ESA operating system that has the ability to process work.

**MVS/JES2.** Multiple Virtual Storage/Job Entry System 2. A z/OS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing.

**MVS/ESA.** Multiple Virtual Storage/Enterprise Systems Architecture.

## N

**NAU.** (1) Network accessible unit. (2) Network addressable unit.

**NCCF.** Network Communications Control Facility.

**NCP.** (1) Network Control Program (IBM licensed program). Its full name is Advanced Communications Function for the Network Control Program. Synonymous with *ACF/NCP*. (2) Network control program (general term).

**NetView.** An IBM licensed program used to monitor a network, manage it, and diagnose network problems. NetView consists of a command facility that includes a presentation service, command processors, automation based on command lists, and a transaction processing structure on which the session monitor, hardware monitor, and terminal access facility (TAF) network management applications are built.

**network accessible unit (NAU).** A logical unit (LU), physical unit (PU), control point (CP), or system services control point (SSCP). It is the origin or the destination of information transmitted by the path control network. Synonymous with *network addressable unit*.

**network addressable unit (NAU).** Synonym for *network accessible unit*.

**NetView automation procedures.** A sequence of commands, packaged as a NetView command list or a command processor written in a high-level language. An automation procedure performs automation functions and runs under the NetView program.

**NetView automation table (AT).** A table against which the NetView program compares incoming messages. A match with an entry triggers the specified response. SA z/OS entries in the NetView automation table trigger an SA z/OS response to target system conditions. Formerly known as the message automation table (MAT).

**NetView Command list language.** An interpretive language unique to NetView that is used to write command lists.

**NetView (NCCF) console.** A 3270-type console for NetView commands and runtime panels for system operations and processor operations.

**NetView Graphic Monitor Facility (NGMF).** Deprecated term for NetView Management Console.

**NetView hardware monitor.** The component of NetView that helps identify network problems, such as hardware, software, and microcode, from a central control point using interactive display techniques. Formerly called *network problem determination application*.

**NetView log.** The log in which NetView records events pertaining to NetView and SA z/OS activities.

**NetView message table.** See *NetView automation table*.

**NetView Management Console (NMC).** A function of the NetView program that provides a graphic, topological presentation of a network that is controlled by the NetView program. It provides the operator different views of a network, multiple levels of graphical detail, and dynamic resource status of the network. This function consists of a series of graphic windows that allows you to manage the network interactively. Formerly known as the NetView Graphic Monitor Facility (NGMF).

**NetView-NetView task (NNT).** The task under which a cross-domain NetView operator session runs. Each NetView program must have a NetView-NetView task to establish one NNT session. See also *operator station task*.

**NetView-NetView Task session.** A session between two NetView programs that runs under a NetView-NetView Task. In SA z/OS, NetView-NetView Task sessions are used for communication between focal point and remote systems.

**NetView paths via logical unit (LU 6.2).** A type of network-accessible port (VTAM connection) that enables end users to gain access to SNA network resources and communicate with each other. LU 6.2 permits communication between processor operations and the workstation.

**network.** (1) An interconnected group of nodes. (2) In data processing, a user application network. See *SNA network*.

**Network Communications Control Facility (NCCF).** The operations control facility for the network. NCCF consists of a presentation service, command processors, automation based on command lists, and a transaction processing structure on which the network management applications NLDM and NPDA are built. NCCF is a precursor to the NetView command facility.

**Network Control Program (NCP).** An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Network Control Program.

**Networking NetView.** In SA z/OS the NetView that performs network management functions, such as managing the configuration of a network. In SA z/OS it is common to also route alerts to the Networking NetView.

**Network Problem Determination Application (NPDA).** An NCCF application that helps you identify network problems, such as hardware, software, and microcode, from a central control point using interactive display methods. The alert manager for the network. The precursor of the NetView hardware monitor.

**NGMF.** Deprecated term for NetView Management Console.

**NGMF focal-point system.** Deprecated term for NMC focal point system.

**NIP.** Nucleus initialization program.

**NMC focal point system.** See *focal point system*

**NMC workstation.** The NMC workstation is the primary way to dynamically monitor SA z/OS systems. From the windows, you see messages, monitor status, view trends, and react to changes before they cause problems for end users. You can use multiple windows to monitor multiple views of the system.

**NNT.** NetView-NetView task.

**notification message.** An SA z/OS message sent to a human notification operator to provide information about significant automation actions. Notification messages are defined using the customization dialogs.

**notification operator.** A NetView console operator who is authorized to receive SA z/OS notification messages. Authorization is made through the customization dialogs.

**NPDA.** Network Problem Determination Application.

**NPDA focal-point system.** See *focal-point system*.

**NTRI.** NCP/token-ring interconnection.

**nucleus initialization program (NIP).** The program that initializes the resident control program; it allows the operator to request last-minute changes to certain options specified during system generation.

## O

**objective value.** An average Workflow or Using value that SA z/OS can calculate for applications from past service data. SA z/OS uses the objective value to calculate warning and alert thresholds when none are explicitly defined.

**OCA.** In SA z/OS, operator console A, the active operator console for a target system. Contrast with *OCB*.

**OCB.** In SA z/OS, operator console B, the backup operator console for a target system. Contrast with *OCA*.

**OCF.** Operations command facility.

**OCF-based processor.** A central processor complex that uses an operations command facility for interacting with human operators or external programs to perform operations management functions on the CPC.

**OPC/A.** Operations Planning and Control/Advanced.

**OPC/ESA.** Operations Planning and Control/Enterprise Systems Architecture.

**operating system (OS).** Software that controls the execution of programs and that may provide services such as resource allocation, scheduling, input/output control, and data management. Although operating systems are predominantly software, partial hardware implementations are possible. (T)

**operations.** The real-time control of a hardware device or software function.

**operations command facility (OCF).** A facility of the central processor complex that accepts and processes operations management commands.

**Operations Planning and Control/Advanced (OPC/A).** A set of IBM licensed programs that automate, plan, and control batch workload. OPC/A analyzes system and workload status and submits jobs accordingly.

**Operations Planning and Control/ESA (OPC/ESA).** A set of IBM licensed programs that automate, plan, and control batch workload. OPC/ESA analyzes system and workload status and submits jobs accordingly. The successor to OPC/A.

**operator.** (1) A person who keeps a system running. (2) A person or program responsible for managing activities controlled by a given piece of software such as z/OS, the NetView program, or IMS. (3) A person who operates a device. (4) In a language statement, the lexical entity that indicates the action to be performed on operands.

**operator console.** (1) A functional unit containing devices that are used for communications between a computer operator and a computer. (T) (2) A display console used for communication between the operator and the system, used primarily to specify information concerning application programs and I/O operations and to monitor system operation. (3) In SA z/OS, a console that displays output from and sends input to the operating system (z/OS, LINUX, VM, VSE). Also called *operating system console*. In the SA z/OS operator commands and configuration dialogs, OC is used to designate a target system operator console.

**operator station task (OST).** The NetView task that establishes and maintains the online session with the network operator. There is one operator station task for each network operator who logs on to the NetView program.

**operator view.** A set of group, system, and resource definitions that are associated together for monitoring purposes. An operator view appears as a graphic display in the graphical interface showing the status of the defined groups, systems, and resources.

**OperatorView entry.** A construct, created with the customization dialogs, used to represent and contain policy for an operator view.

**OS.** Operating system.

**z/OS component.** A part of z/OS that performs a specific z/OS function. In SA z/OS, component refers to entities that are managed by SA z/OS automation.

**z/OS subsystem.** Software products that augment the z/OS operating system. JES and TSO/E are examples of z/OS subsystems. SA z/OS includes automation for some z/OS subsystems.

**z/OS system.** A z/OS image together with its associated hardware, which collectively are often referred to simply as a system, or z/OS system.

**OSA.** I/O operations can display the open system adapter (OSA) channel logical definition, physical attachment, and status. You can configure an OSA channel on or off.

**OST.** Operator station task.

**outbound.** In SA z/OS, messages or commands from the focal-point system to the target system.

**outbound gateway operator.** The automation operator that establishes connections to other systems. The outbound gateway operator handles communications with other systems through a gateway session. The automation operator sends messages, commands, and responses to the inbound gateway operator at the receiving system.

## P

**page.** (1) The portion of a panel that is shown on a display surface at one time. (2) To transfer instructions, data, or both between real storage and external page or auxiliary storage.

**panel.** (1) A formatted display of information that appears on a terminal screen. Panels are full-screen 3270-type displays with a monospaced font, limited color and graphics. (2) By using SA z/OS panels you can see status, type commands on a command line using a keyboard, configure your system, and passthru to other consoles. See also *help panel*. (3) In computer graphics, a display image that defines the locations and characteristics of display fields on a display surface. Contrast with *screen*.

**parallel channels.** Parallel channels operate in either byte (BY) or block (BL) mode. You can change connectivity to a parallel channel operating in block mode.

**parameter.** (1) A variable that is given a constant value for a specified application and that may denote the application. (2) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (3) Data passed to a program or procedure by a user or another program, namely as an operand in a language statement, as an item in a menu, or as a shared data structure.

**partition.** (1) A fixed-size division of storage. (2) In VSE, a division of the virtual address area that is available for program processing. (3) On an IBM Personal Computer fixed disk, one of four possible storage areas of variable size; one can be accessed by DOS, and each of the others may be assigned to another operating system.

**partitionable CPC.** A CPC that can be divided into 2 independent CPCs. See also *physical partition, single-image mode, MP, side*.

**partitioned data set (PDS).** A data set in direct access storage that is divided into partitions, called *members*, each of which can contain a program, part of a program, or data.

**passive monitoring.** In SA z/OS, the receiving of unsolicited messages from z/OS systems and their resources. These messages can prompt updates to resource status displays. See also *active monitoring.*

**PCE.** Processor controller. Also known as the "support processor" or "service processor" in some processor families.

**PDB.** Policy Database

**PDS.** Partitioned data set.

**physical partition.** Part of a CPC that operates as a CPC in its own right, with its own copy of the operating system.

**physical unit (PU).** In SNA, the component that manages and monitors the resources (such as attached links and adjacent link stations) of a node, as requested by a system services control point (SSCP) through an SSCP-PU session. An SSCP activates a session with the physical unit to indirectly manage, through the PU, resources of the node such as attached links.

**physically partitioned (PP) configuration.** A mode of operation that allows a multiprocessor (MP) system to function as two or more independent CPCs having separate power, water, and maintenance boundaries. Contrast with *single-image (SI) configuration*.

**POI.** Program operator interface.

**policy.** The automation and monitoring specifications for an SA z/OS enterprise. See *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

**policy database.** The database where the automation policy is recorded. Also known as the PDB.

**POR.** Power-on reset.

**port.** (1) System hardware to which the I/O devices are attached. (2) On an ESCON switch, a port is an addressable connection. The switch routes data through the ports to the channel or control unit. Each port has a name that can be entered into a switch matrix, and you can use commands to change the switch configuration. (3) An access point (for example, a logical unit) for data entry or exit. (4) A functional unit of a node through which data can enter or leave a data network. (5) In data communication, that part of a data processor that is dedicated to a single data channel for the purpose of receiving data from or transmitting data to one or more external, remote devices. (6) power-on reset (POR) (7) A function that re-initializes all the hardware in a CPC and loads the internal code that enables the CPC to load and run an operating system.

**PP.** Physically partitioned (configuration).

**PPT.** Primary POI task.

**primary host.** The base program at which you enter a command for processing.

**primary POI task (PPT).** The NetView subtask that processes all unsolicited messages received from the VTAM program operator interface (POI) and delivers them to the controlling operator or to the command processor. The PPT also processes the initial command specified to execute when NetView is initialized and timer request commands scheduled to execute under the PPT.

**primary system.** A system is a primary system for an application if the application is normally meant to be running there. SA z/OS starts the application on all the primary systems defined for it.

**problem determination.** The process of determining the source of a problem; for example, a program component, machine failure, telecommunication facilities, user or contractor-installed programs or equipment, environment failure such as a power loss, or user error.

**processor controller.** Hardware that provides support and diagnostic functions for the central processors.

**processor operations.** The part of SA z/OS that monitors and controls processor (hardware) operations. Processor operations provides a connection from a focal-point system to a target system. Through NetView on the focal-point system, processor operations automates operator and system consoles for monitoring and recovering target systems. Also known as ProcOps.

**processor operations control file.** Named by your system programmer, this file contains configuration and customization information. The programmer records the name of this control file in the processor operations file generation panel ISQDPG01.

**Processor Resource/Systems Manager (PR/SM).** The feature that allows the processor to use several operating system images simultaneously and provides logical partitioning capability. See also *LPAR*.

**ProcOps.** Processor operations.

**ProcOps Service Machine (PSM).** The PSM is a CMS user on a VM host system. It runs a CMS multitasking application that serves as ″virtual hardware″ for ProcOps. ProOps communicates via the PSM with the VM guest systems that are defined as target systems within ProcOps.

**product automation.** Automation integrated into the base of SA z/OS for the products DB2, CICS, IMS, OPC (formerly called *features*).

**program to program interface (PPI).** A NetView function that allows user programs to send or receive data buffers from other user programs and to send alerts to the NetView hardware monitor from system and application programs.

**protocol.** In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components.

**proxy resource.** A resource defined like an entry type APL representing a processor operations target system.

**PR/SM.** Processor Resource/Systems Manager.

**PSM.** ProcOps Service Machine.

**PU.** Physical unit.

# R

**remote system.** A system that receives resource status information from an SA z/OS focal-point system. An SA z/OS remote system is defined as part of the same SA z/OS enterprise as the SA z/OS focal-point system to which it is related.

**requester.** A requester is a workstation software, which enables users to log on to a domain, that is, to the server(s) belonging to this domain, and use the resources in this domain. After the log on to a domain, users can access the shared resources and use the processing capability of the server(s). Because the bigger part of shared resources is on the server(s), users can reduce hardware investment.

**resource.** (1) Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs. (2) In NetView, any hardware or software that provides function to the network. (3) In SA z/OS, any z/OS application, z/OS component, job, device, or target system capable of being monitored or automated through SA z/OS.

**Resource Access Control Facility (RACF).** A program that can provide data security for all your resources. RACF protects data from accidental or deliberate unauthorized disclosure, modification, or destruction.

**resource group.** A physically partitionable portion of a processor. Also known as a *side*.

**Resource Monitoring Facility (RMF) Monitor III.** A program that measures and reports on the availability and activity of system hardware and software resources, such as processors, devices, storage, and address spaces. RMF can issue online reports about system performance problems as they occur.

**Resource Object Data Manager (RODM).** A data cache manager designed to support process control and automation applications. RODM provides an in-memory data cache for maintaining real-time data in an address space that is accessible by multiple applications. RODM also allows an application to query an object and receive a rapid response and act on it.

**resource token.** A unique internal identifier of an ESCON resource or resource number of the object in the IODF.

**restart automation.** SA z/OS-provided automation that monitors subsystems to ensure that they are running. If a subsystem fails, SA z/OS attempts to restart it according to the policy in the automation control file.

**Restructured Extended Executor (REXX).** An interpretive language used to write command lists.

**return code.** A code returned from a program used to influence the issuing of subsequent instructions.

**REXX.** Restructured Extended Executor.

**REXX procedure.** A command list written with the Restructured Extended Executor (REXX), which is an interpretive language.

**RMF.** Resource Measurement Facility.

**RODM.** Resource Object Data Manager.

# S

**SAF.** Security Authorization Facility.

**SA IOM.** System Automation for Integrated Operations Management

**SA z/OS.** System Automation for z/OS

**SA z/OS customization dialogs.** An ISPF application through which the SA z/OS policy administrator defines policy for individual z/OS systems and builds automation control data and RODM load function files.

**SA z/OS customization focal point system.** See *focal point system.*

**SA z/OS data model.** The set of objects, classes and entity relationships necessary to support the function of SA z/OS and the NetView automation platform.

**SA z/OS enterprise.** The group of systems and resources defined in the customization dialogs under one enterprise name. An SA z/OS enterprise consists of connected z/OS systems running SA z/OS.

**SA z/OS focal point system.** See *focal point system.*

**SA z/OS policy.** The description of the systems and resources that make up an SA z/OS enterprise, together with their monitoring and automation definitions.

**SA z/OS policy administrator.** The member of the operations staff who is responsible for defining SA z/OS policy.

**SA z/OS satellite.** If you are running two NetViews on an z/OS system to split the automation and networking functions of NetView, it is common to route alerts to the Networking NetView. For SA z/OS to process alerts properly on the Networking NetView, you must install a subset of SA z/OS code, called an *SA z/OS satellite* on the Networking NetView.

**SA z/OS SDF focal point system.** See *focal point system.*

**SCA.** In SA z/OS, system console A, the active system console for a target hardware. Contrast with *SCB.*

**SCB.** In SA z/OS, system console B, the backup system console for a target hardware. Contrast with *SCA.*

**screen.** Deprecated term for display panel.

**screen handler.** In SA z/OS, software that interprets all data to and from a full-screen image of a target system. The interpretation depends on the format of the data on the full-screen image. Every processor and operating system has its own format for the full-screen image. A screen handler controls one PS/2 connection to a target system.

**SDF.** Status Display Facility.

**SDLC.** Synchronous data link control.

**SDSF.** System Display and Search Facility.

**secondary system.** A system is a secondary system for an application if it is defined to automation on that system, but the application is not normally meant to be running there. Secondary systems are systems to which an application can be moved in the event that one or more of its primary systems are unavailable. SA z/OS does not start the application on its secondary systems.

**server.** A server is a workstation that shares resources, which include directories, printers, serial devices, and computing powers.

**service language command (SLC).** The line-oriented command language of processor controllers or service processors.

**service processor (SVP).** The name given to a processor controller on smaller System/370 processors.

**service period.** Service periods allow the users to schedule the availability of applications. A service period is a set of time intervals (service windows), during which an application should be active.

**service threshold.** An SA z/OS policy setting that determines when to notify the operator of deteriorating service for a resource. See also *alert threshold* and *warning threshold*.

**session.** In SNA, a logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header by a pair of network addresses identifying the origin and destination NAUs of any transmissions exchanged during the session.

**session monitor.** The component of the NetView program that collects and correlates session-related data and provides online access to this information. The successor to NLDM.

**shutdown automation.** SA z/OS-provided automation that manages the shutdown process for subsystems by issuing shutdown commands and responding to prompts for additional information.

**side.** A part of a partitionable CPC that can run as a physical partition and is typically referred to as the A-side or the B-side.

**Simple Network Management Protocol (SNMP).** An IP based industry standard protocol to monitor and control resources in an IP network.

**single image.** A processor system capable of being physically partitioned that has not been physically partitioned. Single-image systems can be target hardware processors.

**single-image (SI) mode.** A mode of operation for a multiprocessor (MP) system that allows it to function as one CPC. By definition, a uniprocessor (UP) operates in single-image mode. Contrast with *physically partitioned (PP) configuration*.

**SLC.** Service language command.

**SMP/E.** System Modification Program Extended.

**SNA.** Systems Network Architecture.

**SNA network.** In SNA, the part of a user-application network that conforms to the formats and protocols of systems network architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAUs), boundary function components, and the path control network.

**SNMP.** Simple Network Management Protocol (a TCP/IP protocol). A protocol that allows network management by elements, such as gateways, routers, and hosts. This protocol provides a means of communication between network elements regarding network resources.

**solicited message.** An SA z/OS message that directly responds to a command. Contrast with *unsolicited message*.

**SSCP.** System services control point.

**SSI.** Subsystem interface.

**start automation.** SA z/OS-provided automation that manages and completes the startup process for subsystems. During this process, SA z/OS replies to prompts for additional information, ensures that the startup process completes within specified time limits, notifies the operator of problems, if necessary, and brings subsystems to an UP (or ready) state.

**startup.** The point in time at which a subsystem or application is started.

**status.** The measure of the condition or availability of the resource.

**status focal-point system.** See *focal-point system*.

**status display facility (SDF).** The system operations part of SA z/OS that displays status of resources such as applications, gateways, and write-to-operator messages (WTORs) on dynamic color-coded panels. SDF shows spool usage problems and resource data from multiple systems.

**steady state automation.** The routine monitoring, both for presence and performance, of subsystems, applications, volumes and systems. Steady state automation may respond to messages, performance exceptions and discrepancies between its model of the system and reality.

**structure.** A construct used by z/OS to map and manage storage on a coupling facility. See cache structure, list structure, and lock structure.

**subgroup.** A named set of systems. A subgroup is part of an SA z/OS enterprise definition and is used for monitoring purposes.

**SubGroup entry.** A construct, created with the customization dialogs, used to represent and contain policy for a subgroup.

**subplex.** Situations where the physical sysplex has been divided into subentities, for example, a test sysplex and a production sysplex. This may be done to isolate the test environment from the production environment.

**subsystem.** (1) A secondary or subordinate system, usually capable of operating independent of, or asynchronously with, a controlling system. (2) In SA z/OS, an z/OS application or subsystem defined to SA z/OS.

**subsystem interface.** The z/OS interface over which all messages sent to the z/OS console are broadcast.

**support element.** A hardware unit that provides communications, monitoring, and diagnostic functions to a central processor complex (CPC).

**support processor.** Another name given to a processor controller on smaller System/370 processors; see *service processor*.

**SVP.** Service processor.

**switches.** ESCON directors are electronic units with ports that dynamically switch to route data to I/O devices. The switches are controlled by I/O operations commands that you enter on a workstation.

**switch identifier.** The switch device number (swchdevn), the logical switch number (LSN) and the switch name

**symbolic destination name (SDN).** Used locally at the workstation to relate to the VTAM application name.

**synchronous data link control (SDLC).** A discipline for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. SDLC conforms to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute and High-Level Data Link Control (HDLC) of the International Standards Organization.

**SYSINFO Report.** An RMF report that presents an overview of the system, its workload, and the total number of jobs using resources or delayed for resources.

**SysOps.** System operations.

**sysplex.** A set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components (coupling devices and timers) and software services (couple data sets).

In a sysplex, z/OS provides the coupling services that handle the messages, data, and status for the parts of a multisystem application that has its workload spread across two or more of the connected processors, sysplex timers, coupling facilities, and couple data sets (which contains policy and states for automation).

A Parallel Sysplex is a sysplex that includes a coupling facility.

**sysplex application group.** A sysplex application group is a grouping of applications that can run on any system in a sysplex.

**sysplex couple data set.** A couple data set that contains sysplex-wide data about systems, groups, and members that use XCF services. All z/OS systems in a sysplex must have connectivity to the sysplex couple data set. See also *couple data set*.

**Sysplex Timer.** An IBM unit that synchronizes the time-of-day (TOD) clocks in multiple processors or processor sides. External Time Reference (ETR) is the z/OS generic name for the IBM Sysplex Timer (9037).

**system.** In SA z/OS, system means a focal point system (z/OS) or a target system (MVS, VM, VSE, LINUX, or CF).

**System Automation for Integrated Operations Management.** (1) An outboard automation solution for secure remote access to mainframe/distributed systems. Tivoli System Automation for Integrated Operations Management, previously Tivoli AF/REMOTE, allows users to manage mainframe and distributed systems from any location. (2) The full name for SA IOM.

**System Automation for OS/390.** The full name for SA OS/390, the predecessor to System Automation for z/OS.

**System Automation for z/OS.** The full name for SA z/OS.

**system console.** (1) A console, usually having a keyboard and a display screen, that is used by an operator to control and communicate with a system. (2) A logical device used for the operation and control of hardware functions (for example, IPL, alter/display, and reconfiguration). The system console can be assigned to any of the physical displays attached to a processor controller or support processor. (3) In SA z/OS, the hardware system console for processor controllers or service processors of processors connected using SA z/OS. In the SA z/OS operator commands and configuration dialogs, SC is used to designate the system console for a target hardware processor.

**System Display and Search Facility (SDSF).** An IBM licensed program that provides information about jobs, queues, and printers running under JES2 on a series of panels. Under SA z/OS you can select SDSF from a pull-down menu to see the resources' status, view the z/OS system log, see WTOR messages, and see active jobs on the system.

**System entry.** A construct, created with the customization dialogs, used to represent and contain policy for a system.

**System Modification Program/Extended (SMP/E).** An IBM licensed program that facilitates the process of installing and servicing an z/OS system.

**system operations.** The part of SA z/OS that monitors and controls system operations applications and subsystems such as NetView, SDSF, JES, RMF, TSO, RODM, ACF/VTAM, CICS, IMS, and OPC. Also known as SysOps.

**system services control point (SSCP).** In SNA, the focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Multiple SSCPs, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its domain.

**Systems Network Architecture (SNA).** The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

**System/390 microprocessor cluster.** A configuration that consists of central processor complexes (CPCs) and may have one or more integrated coupling facilities.

# T

**TAF.** Terminal access facility.

**target.** A processor or system monitored and controlled by a focal-point system.

**target control task.** In SA z/OS, target control tasks process commands and send data to target systems and workstations through communications tasks. A target control task (a NetView autotask) is assigned to a target system when the target system is initialized.

**target hardware.** In SA z/OS, the physical hardware on which a target system runs. It can be a single-image or physically partitioned processor. Contrast with *target system*.

**target system.** (1) In a distributed system environment, a system that is monitored and controlled by the focal-point system. Multiple target systems can be controlled by a single focal-point system. (2) In SA z/OS, a computer system attached to the focal-point system for monitoring and control. The definition of a target system includes how remote sessions are established, what hardware is used, and what operating system is used.

**task.** (1) A basic unit of work to be accomplished by a computer. (2) In the NetView environment, an operator station task (logged-on operator), automation operator (autotask), application task, or user task. A NetView task performs work in the NetView environment. All SA z/OS tasks are NetView tasks. See also *communications task*, *message monitor task*, and *target control task*.

**telecommunication line.** Any physical medium, such as a wire or microwave beam, that is used to transmit data.

**terminal access facility (TAF).** (1) A NetView function that allows you to log onto multiple applications either on your system or other systems. You can define TAF sessions in the SA z/OS customization panels so you don't have to set them up each time you want to use them. (2) In NetView, a facility that allows a network operator to control a number of subsystems. In a full-screen or operator control session, operators can control any combination of subsystems simultaneously.

**terminal emulation.** The capability of a microcomputer or personal computer to operate as if it were a particular type of terminal linked to a processing unit to access data.

**threshold.** A value that determines the point at which SA z/OS automation performs a predefined action. See *alert threshold*, *warning threshold*, and *error threshold*.

**time of day (TOD).** Typically refers to the time-of-day clock.

**Time Sharing Option (TSO).** An optional configuration of the operating system that provides conversational time sharing from remote stations. It is an interactive service on z/OS, MVS/ESA, and MVS/XA.

**Time-Sharing Option/Extended (TSO/E).** An option of z/OS that provides conversational timesharing from remote terminals. TSO/E allows a wide variety of users to perform many different kinds of tasks. It can handle short-running applications that use fewer sources as well as long-running applications that require large amounts of resources.

**timers.** A NetView command that issues a command or command processor (list of commands) at a specified time or time interval.

**TOD.** Time of day.

**token ring.** A network with a ring topology that passes tokens from one attaching device to another; for example, the IBM Token-Ring Network product.

**TP.** Transaction program.

**transaction program.** In the VTAM program, a program that performs services related to the processing of a transaction. One or more transaction programs may operate within a VTAM application program that is using the VTAM application program interface (API). In that situation, the transaction program would request services from the applications program using protocols defined by that application program. The application program, in turn, could request services from the VTAM program by issuing the APPCCMD macro instruction.

**transitional automation.** The actions involved in starting and stopping subsystems and applications that have been defined to SA z/OS. This can include issuing commands and responding to messages.

**translating host.** Role played by a host that turns a resource number into a token during a unification process.

**trigger.** Triggers, in combination with events and service periods, are used to control the starting and stopping of applications in a single system or a parallel sysplex.

**TSO.** Time Sharing Option.

**TSO console.** From this 3270-type console you are logged onto TSO or ISPF to use the runtime panels for I/O operations and SA z/OS customization panels.

**TSO/E.** TSO Extensions.

## U

**UCB.** The unit control block; an MVS/ESA data area that represents a device and that is used for allocating devices and controlling I/O operations.

**unsolicited message.** An SA z/OS message that is not a direct response to a command. Contrast with *solicited message*.

**user task.** An application of the NetView program defined in a NetView TASK definition statement.

**Using.** An RMF Monitor III definition. Jobs getting service from hardware resources (processors or devices) are **using** these resources. The use of a resource by an address space can vary from 0% to 100% where 0% indicates no use during a Range period, and 100% indicates that the address space was found using the resource in every sample during that period. See also *Workflow*.

## V

**view.** In the NetView Graphic Monitor Facility, a graphical picture of a network or part of a network. A view consists of nodes connected by links and may also include text and background lines. A view can be displayed, edited, and monitored for status information about network resources.

**Virtual Storage Extended (VSE).** An IBM licensed program whose full name is Virtual Storage Extended/Advanced Function. It is an operating system that controls the execution of programs.

**Virtual Telecommunications Access Method (VTAM).** An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Virtual Telecommunications Access Method. Synonymous with *ACF/VTAM*.

**VM/ESA.** Virtual Machine/Enterprise Systems Architecture.

**VM Second Level Systems Support.** With this function, Processor Operations is able to control VM second level systems (VM guest systems) in the same way that it controls systems running on real hardware.

**volume.** A direct access storage device (DASD) volume or a tape volume that serves a system in an SA z/OS enterprise.

**volume entry.** A construct, created with the customization dialogs, used to represent and contain policy for a volume.

**volume group.** A named set of volumes. A volume group is part of a system definition and is used for monitoring purposes.

**volume group entry.** An construct, created with the customization dialogs, used to represent and contain policy for a volume group.

**Volume Workflow.** The SA z/OS Volume Workflow variable is derived from the RMF Resource Workflow definition, and is used to measure the performance of volumes. SA z/OS calculates Volume Workflow using:

```
                  accumulated
                     Using
Volume     =  ------------------------  * 100
Workflow %    accumulated + accumulated
                Using         Delay
```

The definition of **Using** is the percentage of time when a job has had a request accepted by a channel for the volume, but the request is not yet complete.

The definition of **Delay** is the delay that waiting jobs experience because of contention for the volume. See also *Address Space Workflow*.

**VSE.** Virtual Storage Extended.

**VTAM.** Virtual Telecommunications Access Method.

# W

**warning threshold.** An application or volume service value that determines the level at which SA z/OS changes the associated icon in the graphical interface to the warning color. See *alert threshold*.

**workflow.** See *Address Space Workflow* and *Volume Workflow*.

**workstation.** In SA z/OS workstation means the *graphic workstation* that an operator uses for day-to-day operations.

**write-to-operator (WTO).** A request to send a message to an operator at the z/OS operator console. This request is made by an application and is handled by the WTO processor, which is part of the z/OS supervisor program.

**write-to-operator-with-reply (WTOR).** A request to send a message to an operator at the z/OS operator console that requires a response from the operator. This request is made by an application and is handled by the WTO processor, which is part of the z/OS supervisor program.

**WTO.** Write-to-Operator.

**WTOR.** Write-to-Operator-with-Reply.

**WWV.** The US National Institute of Standards and Technology (NIST) radio station that provides standard time information. A second station, known as WWVB, provides standard time information at a different frequency.

# X

**XCF.** Cross-system coupling facility.

**XCF couple data set.** The name for the sysplex couple data set prior to MVS/ESA System Product Version 5 Release 1. See also *sysplex couple data set*.

**XCF group.** A set of related members that a multisystem application defines to XCF. A member is a specific function, or instance, of the application. A member resides on one system and can communicate with other members of the same group across the sysplex.

**XRF.** Extended recovery facility.

# Numerics

**390-CMOS.** Processor family group designator used in the SA z/OS processor operations documentation and in the online help to identify any of the following S/390 CMOS processor machine types: 9672, 9674, 2003, 3000, or 7060. SA z/OS processor operations uses the OCF facility of these processors to perform operations management functions. See *OCF-based processor*.

# Index

# Readers' Comments — We'd Like to Hear from You

**System Automation for z/OS**
**Planning and Installation**
**Version 3 Release 2**

**Publication No. SC33-8261-04**

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:
- Send your comments to the address on the reverse side of this form.
- Send a fax to the following number: FAX (Germany): 07031+16-3456
  FAX (Other Countries): (+49)+7031-16-3456
- Send your comments via e-mail to: s390id@de.ibm.com

If you would like a response from IBM, please fill in the following information:

| | |
|---|---|
| Name | Address |
| Company or Organization | |
| Phone No. | E-mail address |

## Readers' Comments — We'd Like to Hear from You

SC33-8261-04

IBM®

Cut or Fold
Along Line

Fold and Tape    **Please do not staple**    Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

# BUSINESS REPLY MAIL

FIRST-CLASS MAIL    PERMIT NO. 40    ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

Fold and Tape    **Please do not staple**    Fold and Tape

Cut or Fold
Along Line

Program Number: 5698-SA3

Printed in USA

SC33-8261-04