IBM Z System Automation
4.4

Planning and Installation

SC34-2716-04

Note

Before using this information and the product it supports, read the information in Appendix H, “Notices,” on page 227.

Edition Notes

This edition applies to IBM Z® System Automation (Program Number 5698-SA4) Version 4 Release 4, an IBM licensed program, and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC34-2716-03.

Contents

Tables

Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully. IBM Z System Automation supports several user interfaces. Product functionality and accessibility features vary according to the interface.

The major accessibility features in this product enable users in the following ways:

- Use assistive technologies such as screen reader software and digital speech synthesizer, to hear what is displayed on screen. Consult the product documentation of the assistive technology for details on using those technologies with this product and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Magnify what is displayed on screen.

The product documentation includes the following features to aid accessibility:

- All documentation is available to both HTML and convertible PDF formats to give the maximum opportunity for users to apply screen-reader software
- All images in the documentation are provided with alternative text so that users with vision impairments can understand the contents of the images.

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 *ISPF User's Guide* 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.

How to send your comments to IBM

We appreciate your input on this publication. Feel free to send us any comments you might have.

If you have feedback to the manuals

If you have comments on the manuals, like clarity, accuracy, and completeness of the information, use the Feedback channel on IBM Documentation to send your comments.

1. Click **Feedback** at the upper right corner on this page of IBM Z System Automation documentation.
2. Choose one mail application and log in or log in to the mail application that's invoked by default. A draft email is displayed after login.
3. In the email body, write down your feedback. Please include the specific book and topic name that you're commenting on.
4. Send the email to the default recipient.
5. IBM Z System Automation team will respond to you by email as soon as possible.

If you have a technical problem

Use one of the following feedback methods:

- Contact your IBM service representative
- Call IBM technical support

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 submit.

About this publication

This publication describes IBM Z System Automation (SA z/OS, or System Automation for short) 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 Publication

This information is intended primarily for system programmers and automation administrators 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 Z NetView (formerly called IBM Tivoli NetView for z/OS) .

Where to Find More Information

Z System Automation Library

Table 1 on page xvii shows the information units in Z System Automation library. These manuals can be downloaded from IBM Documentation.

Table 1. Z System Automation library

Title	Form Number	Description
Get Started Guide	SC27-9532	This book is intended for SA z/OS beginners. It contains the information about early planning, configuring the product, making it secure, customizing your automation environment, and the basic operational tasks that you perform on a daily basis.
Planning and Installation	SC34-2716	Describes SA z/OS new capabilities and how to plan, install, configure, and migrate SA z/OS.
Customizing and Programming	SC34-2715	Describes how to adapt the standard installation, add new applications to automation, write your own automation procedures, monitor applications, enable alerting, and more.
Defining Automation Policy	SC34-2717	Describes how to define and maintain the automation policy.
User's Guide	SC34-2718	Describes SA z/OS functions and how to use SA z/OS to monitor and control systems.
Messages and Codes	SC34-2719	Describes the problem determination information of SA z/OS, including messages, return codes, reason codes, and status codes.

Table 1. Z System Automation library (continued)		
Title	**Form Number**	**Description**
Operator's Commands	SC34-2720	Describes the operator commands available with SA z/OS, including their purpose, format, and specifics of how to use them.
Programmer's Reference	SC34-2748	Describes the programming interfaces of SA z/OS and the definitions for the status display facility (SDF).
End-to-End Automation	SC34-2750	Describes the end-to-end automation adapter for z/OS and how it enables end-to-end automation and how it connects to Automation Dashboards for Z Automation Web Console.
Product Automation Programmer's Reference and Operator's Guide	SC34-2714	Describes how to customize and operate product automation components (CICS, Db2, and IMS automation) with SA z/OS to provide a simple and consistent way to monitor and control all of the CICS, Db2, and IMS regions, both local and remote, within your organization.
Workload Scheduler Programmer's Reference and Operator's Guide	SC34-2749	Describes how to customize and operate ZWS/TWS Automation.

Related Product Information

For information that supports Z System Automation, visit the z/OS library in IBM Documentation:

https://www.ibm.com/docs/en/zos

Summary of Changes for SC34-2716-03

This document contains information previously presented in IBM Z System Automation 4.2.0 Planning and Installation, SC34-2716-02.

Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Migration to Java 17 (OA65704)

Starting from APAR OA65704, Java Runtime Environment (JRE) 17 or higher is required to use the components of System Automation Operations REST Server, System Automation Info Broker, and Policy Services Provider. The previously supported minimum Java JRE 1.8 is going to sunset. If you have used any of these components with an older Java version, proceed with the migration steps in the following list.

- Operations REST Server: Step E: Migrate to Java 17 / Java 21 (OA65704).
- System Automation Info Broker: Step G: Migrate to Java 17 / Java 21 (OA65704).
- Policy Services Provider

Gateway limitation

Added the gateway limitation for the case when multiple IBM Z System Automation instances run on one system. See “System Operations Considerations” on page 27.

PassTicket support (OA64126)

With APAR OA64126, IBM Z System Automation has been enhanced to exploit PassTickets to authenticate the user that attempts to log in to OMEGAMON or make a SOAP request. This feature avoids storing password credentials in clear or managing the passwords by using the System Automation's password data set.

Before you can use the PassTicket function, complete the setup as described in “Authentication using PassTickets” on page 178.

New security considerations for processor hardware connections

Inside IBM Z System Automation, TLS itself is not yet supported for any hardware automation connection. But you can use the hybrid SNMP connection (ISQET320) or the INTERNAL connection. Since ISQET32 or INTERNAL communication is kept inside IBM Z hardware, which includes the IBM Z hardware network, there is no need for additional transport security such as TLS to secure it in a public network environment. Ensure that you're aware of the security considerations in topic “Processor Hardware Connection Security Considerations” on page 183.

OA63121

System Automation implements IBM Db2 for z/OS as an alternative option to store dynamic resources. For configuration, see “Step 28: Configure Db2 as an alternative database of dynamic resources” on page 123.

Generally available (GA) level

A new optional component that is called Policy Services Provider is introduced in Z System Automation 4.3 that allows you to generate the PDB report in JSON format. For configuration, see “Step 30: Configure the Policy Services Provider” on page 135

Part 1. Planning

This information provides details on the following:

Chapter 1. SA z/OS Prerequisites and Supported Equipment

SA z/OS Components

SA z/OS consists of the following components:

- System operations (*SysOps* for short)
- Processor operations (*ProcOps* for short)

Refer to "Component Description" on page 19 for details.

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

- CICS®
- Db2®
- IMS
- ZWS

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 IBM Z NetView 6.4.0. and later.

SA z/OS System Operations

The System Operations base program can run on any processor supported by IBM Z NetView 6.4.0 and later, and z/OS 2.3 and later.

Functional Prerequisites

The processor hardware interfaces of SA z/OS support the following processor hardware family:

- IBM Z (IBM z13®/z13s® or later)

The minimum Support Element (SE) and Hardware Management Console (HMC) workplace versions that are required by the processor hardware interfaces of SA z/OSare as follows:

- SE: Workplace Version 2.13.1
- HMC: Workplace Version 2.13.1

With SE/HMC Workplace Version 2.13.1, a new IML mode for CPCs, IBM Dynamic Partition Manager (DPM) is available. The following additional prerequisites apply:

- SE: Workplace Version 2.13.1

  The attached CPC must run in PR/SM mode. The DPM mode is not supported by the SA z/OS processor hardware interfaces (ProcOps, SA-BCPii).

- HMC: Workplace Version 2.13.1

  At least one of the defined CPCs in the Defined CPC Group must run in PR/SM mode. Otherwise, this HMC cannot be used to target processors or systems using a ProcOps connection to this HMC.

  At least one of the defined CPCs in the Defined CPC Group must run in PR/SM mode. Otherwise, the SA-BCPii requests routing function is not operational.

The processor hardware interfaces of SA z/OS provide limited support for zEnterprise EC12 (zEC12) and IBM zEnterprise BC12 (zBC12) processors. SA-BCPii (ISQET32) connections or ProcOps SNMP connections to the SE of an zEC12 or zBC12 processor are not supported. SA z/OS allows to monitor and control this processor hardware with ProcOps using an SNMP HMC connection only.

For a zEC12 or zBC12 processor, the required HMC workplace versions that are supported by SA z/OS are:

- HMC: Workplace Version 2.13.1 or 2.14.1

Later HMC workplace versions no longer provide support for zEC12 and zBC12 processor hardware.

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 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, V2.3 or later.
IBM Z NetView, V6.3.0 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 runtime 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:	
HTML browser	For customization reports
z/VM® 7.1	For VM Second Level Systems support
IBM Tivoli® OMEGAMON® XE on z/OS V5.3 IBM Tivoli OMEGAMON XE for CICS on z/OS V5.3 IBM Tivoli OMEGAMON XE for IMS on z/OS V5.3 IBM Tivoli OMEGAMON XE for DB2® Performance Expert on z/OS V5.3	For the following commands: • INGMTRAP • INGOMX
IBM CICS Transaction Server for z/OS V5.4, or later	For integrated automation of CICS address spaces and CICSPlex®-based monitoring
IBM Db2 for z/OS V12.1, or later	For integrated automation of Db2 address spaces
IBM IMS V15.1, or later	For integrated automation of IMS address spaces
IBM Workload Scheduler for z/OS V9.3, or later	For integrated automation of IWS/ZWS address spaces
Java™ Runtime Environment (JRE) 1.8 (32 bit) installed on z/OS	For using the System Automation data store, E2E automation adapter and E2E automation agent
Java Runtime Environment (JRE) 17 or higher installed on z/OS	For using the System Automation Operations REST Server, System Automation Info Broker, and Policy Services Provider
IBM Z NetView V6.3.0, or later	For using the System Automation Operations REST Server
Workstation Prerequisites:	
IBM Tivoli Business Service Manager for z/OS V6.2	For event notification
IBM Tivoli Netcool/OMNIbus V8.1, or later	For event notification

Supported Hardware

SA z/OS processor operations supports monitoring and control functions for the processors of the following IBM mainframe family, including the logical partitioning of these processors:

- IBM Z (IBM z13/z13s or later)

Operator Terminals

SA z/OS supports any display supported by ISPF V6.1 or higher. This is required for access to the SA z/OS customization dialogs.

The SA z/OS customization dialogs must be used with a terminal type of 3278.

Supported Operating Systems

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

- z/OS
- z/VM
- z/VSE
- Linux on z Systems

Chapter 2. What's New in 4.4.0

This information contains an overview of the major changes to IBM Z System Automation 4.3.0. Use this information to check the impact on your user-written programming interfaces, such as automation procedures.

What's New (GA-level)

This topic lists all the enhancements that are made to IBM Z System Automation 4.4.0 when it was generally available (GA) on October 24, 2025 .

- **Inbound System Automation Operations API enhancements**

  The Inbound SA Operations API is enhanced to support Application Groups (APG). It sends event notifications when resource actions such as start and stop are completed. It also provides additional functions for managing APGs, including notifications through Server-Sent Events (SSE) for actions like start, stop, and policy activation. Information endpoints are available for detailed status updates.

- **Inbound System Automation Customization API enhancements**

  The Inbound System Automation Customization API uses the SA Policy Service Provider to manage Timer (TMR) entries within an SA policy. It also creates and retrieves policy reports in JSON format to support integration and analysis.

- **Addition of Outbound System Automation REST client callable module**

  The Outbound SA REST client callable module is added to help ease the creation of NetView and SA REXX scripts that call any RESTful API to perform GET or POST operations on API endpoints.

- **Enhanced System Automation plug-in for Zowe CLI**

  Updated and enhanced versions of the SA plug-in for Zowe CLI use the capabilities and functions that are provided by the SA Operations API to improve integration and automation through the plug-in.

- **Sample policy added for IBM Z Container Support (zOSCP)**

  System Automation provides a sample policy for IBM Z Container Support (zOSCP), also known as Kubernetes that automates infrastructure that runs pods with containerized workloads.

- **Support for 64-bit Java environments and Java version 21**

  System Automation supports 64-bit Java environments. Most SA Java components support Java version 21, except for the SA E2E Automation Adapter, which requires a different Java version to communicate with the IBM Z Automation Web Console.

- **INGREST method for REST API access**

  The generic callable method INGREST enables NetView and SA REXX scripts to use any standard REST API.

- **Z Automation Web Console dashboard enhancements**

  Z Automation Web Console dashboard adds hardware management views by using data from SA ProcOps and provides visibility into hardware status.

- **System Automation Policy Customization enhancements**

  System Automation includes enhancements through the SA Policy Service Provider and the SA Customization API. You can create ACF fragments directly from the SA Customization Dialog. The customization REST API enables you to add, modify, and delete Timer entries in SA policies to support maintenance and migration tasks. Additionally, you can create and retrieve policy reports in JSON format.

Agentic AI prototypes use the SA Customization API to interact with policy elements.

- **New best practice sample policies**

  System Automation now includes best practice sample policies for the z/OS Container Platform (zOSCP) and the z/OS Control Plane Appliance (zCPA). These additions support streamlined policy setup and provide guidance for orchestrating containerized workloads on z/OS.

- **Support mixed case for Startup parameters**

  Startup parameters can now also be entered in mixed case on application classes, which can be inherited by the downward classes or applications. This mixed case support eliminates your need to define mixed case on instance level. For more information, see APPLICATION INFO. Also, note that the file update function does not support mixed case characters when you import Application INFO policy item. For more details of this restriction, see Supported Policies.

- **OPER change in** INGEXEC

  OA66074 I - SA410 - INGEXEC honors automated function in OPER= parameter. See INGEXEC.

- **Validated boot support to ACTIVATE and LOAD commands**

  The ACTIVATE and LOAD commands now support validated boot through three optional parameters: LDIPL, SECURE, and TYPE. These parameters enable secure and auditable Initial Program Load (IPL) operations when used with LOAD_STD in ACTIVATE or IPLADDR in LOAD. See ACTIVATE and LOAD.

- **Processor Operations support for Hardware Console Availability Controls**

  Starting with the availability of the IBM z16® mainframe models, Hardware Console Availability Controls (HWCAC) are introduced to inform about an upcoming outage of the Hardware Console as well as the end of the outage.

  Processor Operations can receive those events and provides out-of-the-box automation and integration with user automation through an updated add-on policy.

- **Discontinuation of Processor Operations Service Machine (PSM)**

  Starting with this release, the Processor Operations Service Machine (PSM) and all associated commands are discontinued and no longer supported in Z System Automation.

System Automation Operations REST API V1.1

A new version V1.1 of the is available as part of the Z System Automation 4.3 release (For 4.2 release, the API V1.1 is delivered through APAR OA61704). Compared with the previous version V1.0, additional endpoints are added to the API that now allows you to get details of requests and delete them, and work with automation agents and manager including activating or refreshing the automation policy.

The current version of the System Automation Operations REST API has been tested and certified for the Zowe™ API Mediation Layer V1 LTS (Long Time Support) Version. For more information about the "Zowe Conformance Program", see Zowe Conformance Program - Open Mainframe Project™.

IBM Z System Automation Plug-in for Zowe CLI

A plug-in for Zowe CLI is introduced in Z System Automation 4.3, which you can use to interact with Z System Automation from your preferred command line on Microsoft™ Windows™, Linux®, or Apple® Macintosh® operating systems. This "System Automation Plug-in for Zowe" allows you to contact your SAPlex through the "System Automation Operations REST API", for example, for the following services:

- List and manage (start or stop, suspend or resume) defined resources.
- List, create, or delete dynamic resources.
- List, create, or delete requests.
- Refresh automation policies.
- Work with automation agents and automation managers.

In order to use this plug-in with your System Automation systems, you have to configure and run the optional component "System Automation Operations REST Server" that provides the Operations REST API. The plug-in still works with the older API version V1.0. However, many commands require the latest version V1.1, which is part of Z System Automation 4.3 release.

The current version for the System Automation Plug-in for Zowe has been tested and certified for the Zowe CLI V1 LTS (Long Time Support) Version. For more information about the "Zowe Conformance Program", see Zowe Conformance Program - Open Mainframe Project.

A System Automation Plug-in for Zowe has been made available already for System Automation 4.2 in 2021 as technical preview service on npmjs.com (https://www.npmjs.com/package/@ibm/system-automation-for-zowe-cli). Since this release, a full supported version of this plug-in is part of the product and shipped as a package in the zFS file system. You can transfer it from there to any supported client system (Windows, Linux, Mac) that has the Zowe CLI installed and contact the System Automation Operations REST API of your System Automation systems. For more information, see Using the System Automation Plug-in for Zowe CLI to Operate System Automation.

Processor Operations (ProcOps) Enhancements

- **New HOLD mode of ISQET32 sessions**

  You can now put an ISQET32 session in the new HOLD mode that keeps the session alive, as an alternative to the existing SUSPENDED mode that closes an active session. You can use this new HOLD mode to temporarily halt any hardware SA-BCPii communication and automation to IBM Z processors, for example, for a planned Support Element outage. In this mode, commands cannot be sent to the hardware, nor will events from the hardware be routed to operators or autotasks for processing. For more information about the HOLD mode and how to hold and resume a session, see Using the ProcOps HOLD Session Mode in the *User Guide* manual.

- **System Recovery Boost support**

  System Recovery Boost (SRB) indicators are added to the ProcOps data model, API, and UIs. In addition to the ISQCCMD GETIBOOST STATUS and EVALUATE options, the ISQVARS target attention (TATTN) keyword new value "BOOST" now informs automation applications and operators when ProcOps monitoring detects system running in LPARs with BOOST mode on. For more information about how System Recovery Support is supported by Z System Automation, see IBM Z System Recovery Boost in the *User Guide* manual.

System Automation Info Broker

To forward SDF messages into Service Management Unite (SMU) Automation, where the operators can easily monitor SDF messages, delete captured messages, or reply to WTOR messages in the modern UI, a new optional component called System Automation Info Broker is introduced in System Automation 4.3. The main idea of the System Automation Info Broker component is to provide an infrastructure that can publish System Automation event messages to a central repository, that is an Apache Kafka system, where the event messages can be consumed by different consumers, for example, some modern display facilities. The first step of this component is to forward SDF messages to SMU via Apache Kafka. The forwarded SDF messages are placed as JSON document into the Kafka topic, where it can be consumed by SMU and displayed on the SMU "SDF Messages Overview" dashboard. For the use case in SMU, see the new features added to SMU.

Except for SDF messages, you can also forward any user-defined messages via the System Automation Info Broker component by using the new command INGIBRKR.

Note that a running "Apache Kafka" server is a prerequisite for this feature and is not delivered with Z System Automation. For more information about how to configure this component, see Configure System Automation Info Broker.

Service Management Unite Automation

The following major changes are made to Service Management Unite Automation V1.1.9:

- **New dashboard - "SDF Messages Overview"**

  With the new dashboard for SDF messages, you can easily view and operate all captured messages and Write to Operator with Reply (WTOR) messages in your environment on a single dashboard. Service Management Unite leverages the new System Automation Info Broker, included in IBM Z System Automation 4.3, which streams SDF messages to Service Management Unite via Apache Kafka for near real-time dashboard updates. You can monitor the priority of the messages at a glance and delete the captured messages or reply to the WTOR messages directly from the dashboard.

- **New dashboard - "Manage Dynamic Resources"**

  The new dashboard for dynamic resources management allows you to dynamically create applications that are automated by Z System Automation based on pre-defined templates. You can view the dynamic resources and the available templates. In addition, you can easily create dynamic resources through a guided workflow and resume dynamic resources.

- **Miscellaneous enhancements and fixes**

  – The "Request Start" function is enhanced. You can now specify the start type when you send a request to start automation resources. Valid start types are `NORM` (default) and any other types that are defined in the policy database.

  – The status of the System Automation agent is mapped to a richer set of status icons so you can easily distinguish important status information. For more information, see How to: Customize the widget to show the SA agent status.

  – The IBM Service Management Unite console URL is customized and enhanced when the port number of the console is set to 443.

Miscellaneous Capabilities

- **IBM Z Workload Scheduler (ZWS) Automation**

  To save operations time, System Automation infrastructure is enhanced to provide the ability of an immediate result check right after the termination of an asynchronous command that is routed from ZWS to System Automation. The checking routine is invoked immediately after the command processing finished, without waiting until the max wait time is reached. The maximum wait time value (within ZWS) should be specified with a prefixed 'I' to indicate immediate checking. For more information, see User-supplied Completion Checking Routine.

- **Job Log Monitoring**

  Job Log Monitoring capability is enhanced to monitor temporary data sets and automate applications based on the contents of such data sets. For more information, see Job Log Monitoring Overview and Access to JES Spool Output Data Sets.

- **Force prepare move for SERVER application groups**

  To improve high availability when recycling SERVER application groups, you can now force the prepare move function for SERVER APGs during rolling recycle only. To use this feature, set the new option **Force prepare move on rolling recycle** to YES on the group's Application Group Information policy. With this setting, a member will not be stopped until the started replacement is available during rolling recycle and hence improves the group's high availability.

- **Transition of LPAR management into ProcOps component**

  Starting from Z System Automation 4.3, the LPAR management function is transited to the ProcOps component. The previous LPAR management function uses the SA-BCPii INTERNAL protocol and only a subset of the ProcOps common commands are usable. After this transition, the LPAR management capabilities retain the same, just using a different hybrid SNMP protocol, and the full set of ProcOps common commands are usable. For more information about the migration steps, see Migrating from LPAR Management to ProcOps.

- **Hardware validation function disabled by default**

  The default setting of the advanced automation CGlobal variable AOF_AAO_HW_VALIDATION is changed from YES to NO.

Continuous Enhancements (post-GA service-level)

This topic lists all the new functions or enhancements that are incorporated into IBM Z System Automation 4.4.0, since it was generally available (GA) on October 24, 2025.

OA65704 – Migration to Java 17 (April 2024)

Starting from APAR OA65704, Java Runtime Environment (JRE) 17 or higher is required to use the components of System Automation Operations REST Server, System Automation Info Broker, and Policy Services Provider. If you have used any of these components with an older Java version, proceed with the migration steps in *Planning and Installation*.

Preview the functions of System Automation Operations REST API (April 2024)

The Operations REST Server component of IBM Z System Automation provides an API that allows developers of other programs or products to integrate System Automation operations as part of their workflow. Detailed documentation of the Operations REST API is accessible through a built-in Swagger UI, which becomes available after the Operations REST Server component is started.

For users who want to preview the API functions without installing the required component, a copy of the API documentation is added. However, it's for preview purpose only. If you want to execute the endpoints or integrate with other products, see "Step 29: Configure and Run the System Automation Operations REST Server" on page 125.

OA66029 – Enhancement to dynamic resource documentation (Feb 2024)

Automation managers and agents uniquely identify static resources (created in the policy) and dynamic resources (created with INGDYN) by their names. You should avoid name conflicts of static and dynamic resources. In case you want to turn a dynamic resource to static, you need to manually delete the dynamic resource. The reasons to do so and how to do it are explained in "Dynamic resources" in *IBM Z System Automation User's Guide*.

OA65978 – Enhancement to policy browse mode (Jan 2024)

Previously in IBM Z System Automation 4.3, when you were browsing the policy and then used an entry type as a fast path (such as =APL), the policy was switched to the edit mode without any warning.

To avoid that you might accidentally change the policy when you assume that you're browsing it, the handling of the primary panel and commands on the Customization Dialog is changed in APAR OA65978. If you want to switch from browse mode to edit mode, you need to explicitly add the edit option (1, E, or EDIT). A notification message is also added on the panel to indicate that the policy has been switched to edit mode.

For more information, see "How to Use an Entry Type as a Fast Path" in *IBM Z System Automation Defining Automation Policy*.

Service Management Unite V1.1.11.0 (Dec 2023)

You can now view your IBM Z System Automation policy reports and refresh System Automation policies through IBM Service Management Unite.

- Support for viewing System Automation policy database reports:

  A new **Manage IBM Z System Automation Policies** dashboard allows you to refresh policy databases and view detailed policy reports.
- Support for refreshing System Automation policies:

  You can refresh the System Automation policy configuration with the **Refresh Policy Configuration** operation through Service Management Unite.

See What's new in V1.1.11.0 for more information.

Definition of group recovery mode (Nov 23)

To learn how IBM Z System Automation recovers from application or system failures and when an application group enters and remains in the recovery mode, see "Group Recovery Mode" in *User's Guide*.

OA64126 – PassTicket support for INGOMX (Nov 23)

With APAR OA64126, IBM Z System Automation has enhanced the INGOMX command to exploit PassTickets to authenticate the user that attempts to log in to OMEGAMON or make a SOAP request. This feature avoids storing password credentials in clear or managing the passwords by using the System Automation's password data set.

To use the PassTicket function, first complete the setup as described in “Authentication using PassTickets” on page 178.

To use PassTickets for OMEGAMON classic sessions, specify the SAF application ID and the new keyword PTKT as the password in the OMEGAMON SESSIONS Policy Item.

To use PassTickets for SOAP requests, specify the SAF application ID and the new keyword PTKT as the password in the SOAP SERVER Policy Item if the SOAP server is defined in the policy. If the SOAP server is not defined in the policy, specify the credentials in the INGOMX command line.

Service Management Unite V1.1.10.1 (September 2023)

This information contains an overview of the major changes in IBM Service Management Unite Automation V1.1.10.1.

- SA z/OS request dialog enhancements:
  - Added the **Ignore suspended automation** option for the SA z/OS request dialog.
  - Added the **Only Children** option for the SA z/OS stop/restart request dialog.
- Added support for special characters in dynamic resource template names.

Processor hardware connection security considerations (July 2023)

If your organization requires TLS-level security for hardware automation connections, ensure that you're aware of these security considerations.

OA64366 – Updated version of System Automation Plug-in for Zowe CLI (Mar 2023)

OA64366 includes an updated version of System Automation Plug-in for Zowe CLI. This plug-in contains the following changes compared to its prior version:

- This plug-in supports and requires the current version of Zowe CLI, LTS Version 2. It cannot be installed on Zowe CLI LTS Version 1. For more information about how to install or update Zowe CLI, see Zowe documentation.
- The plug-in now supports the new way of configuration not using the deprecated profiles anymore. Credentials are now stored securely in the Zowe CLI built-in credential store. No additional installed credential store plug-in is required any longer. For more information, see Initializing team configuration in Zowe documentation.
- The **`zowe sa list templates`** command now supports a new option **`--showinstances`**. As shown in this example, you can use this option to easily list all dynamic resource instances that are created based on a specific template.

```
PS C:\user\0123456789> zowe sa ls tmp --name=T_NONMVS --showinstances
name              timestamp           status    Comment
DYNTST1/APL/SYS1 2023-03-13T17:43:24 OK
DYNTST2/APL/SYS1 2023-03-14T14:08:04 OK
```

For more information about how to install this plug-in, see Install System Automation Plug-in for Zowe CLI.

Service Management Unite V1.1.10 (March 2023)

You can now disable the Manage Schedules function based on your needs. To disable the Manage Schedule function, you must enter the SMU container, set the `enable-manage-schedule=false` in `cfg/eez.enable.properties`. The change takes effect immediately and you do not need to restart SMU container. See How to: Manage automation schedules for more information.

OA62518 – Enhancements to System Automation Data Store, Info Broker, and Operations REST API (Feb 2023)

OA62518 delivers the following enhancements to IBM Z System Automation 4.3.0. In order to use any of these new features, you must use and adapt the latest configuration files.

- **Changes for System Automation Data Store**
  - Added support for user-specific JVM options in the '`ing.datastore.environment`' file.

```
# Add optional user specified java options
USR_JOPT="$USR_JOPT -Xms64m"
USR_JOPT="$USR_JOPT -Xmx512m"
```

- Added system symbol support for environment setup (`ing.datastore.environment`) and all properties available at JVM runtime (`ing.datastore.properties`). To enable this feature, set the following new property in the '`ing.datastore.environment`' file to true. For more information, see System Symbol Support for System Automation Data Store.

```
# Enable system symbol support. [true/false]
ENABLE_SYSTEM_SYMBOL_SUPPORT=true
```

- **Changes for System Automation Info Broker**
  - Added support for user-specific JVM options in the '`ing.infobroker.environment`' file.

```
# Add optional user specified java options
USR_JOPT="$USR_JOPT -Xms64m"
USR_JOPT="$USR_JOPT -Xmx512m"
```

  - Added system symbol support for environment setup (`ing.infobroker.environment`) and all properties available at JVM runtime (`ing.infobroker.properties`, `ing.infobroker.security.properties`). To enable this feature, navigate to the file '`ing.infobroker.environment`' and set the following new property to true. For more information, see System Symbol Support for System Automation Info Broker.

```
# Enable system symbol support [true/false].
ENABLE_SYSTEM_SYMBOL_SUPPORT=true
```

- **Changes for System Automation Operations REST Server**
  - RACF key rings are now supported by default. Support for ICSF (z/OS Integrated Cryptographic Service Facility) key rings can be enabled with the new option in the 'ing.operations.environment' file. For more information, see Configure the properties for using RACF or ICSF key rings.

```
# Enable optional key ring support
ENABLE_KEYRING_SUPPORT=true

# IBM's crypto provider for RACF key ring
IBM_CRYPTO_PROVIDER=”com.ibm.crypto.provider”
# IBM's crypto provider for ICSF key ring
# IBM_CRYPTO_PROVIDER=”com.ibm.crypto.hdwrCCA.provider”
```

  - Added support for user-specific JVM options in the '`ing.operations.environment`' file.

```
# Add optional user specified java options
USR_JOPT="$USR_JOPT -Xms64m"
USR_JOPT="$USR_JOPT -Xmx512m"
```

  - Added system symbol support for environment setup (`ing.operations.environment`) and all properties available at JVM runtime (`ing.operations.properties`, `ing.operations.security.properties`). For more information, see System Symbol Support for Operations REST Server.
  - Added an endpoint "`GET /templates/{templateId}/instances`" to list all the dynamic resource instances or those that are created based on a specific template.
  - Added a security feature Rate Limiter, which controls the maximum number of calls you want to allow in a particular time interval for the Operations REST API. For more information, see Rate Limiter.

    Note: Rate Limiter is a breaking change as it can cause issues for client applications that do not handle the appropriate HTTP error code.

OA64276 – Processor Operations returns additional IBM z16 System Recovery Boost information (Feb 2023)

Additional System Recovery Boost attributes were introduced in IBM z16®, for example, total boost time or remaining boost time for the CPC image. This APAR OA64276 enables IBM Z System Automation Processor Operations to return those additional attributes.

You can use the STATUS option of the GETIBOOST common command to query System Recovery Boost information. If the target hardware supports the additional System Recovery Boost attributes, they are returned as part of the ISQ900I system console message.

OA64035 – Processor Operations enhancements (Dec 2022)

This APAR OA64035 enables two new features for IBM z16 in IBM Z System Automation Processor Operations:

- Support for IBM Z Flexible Capacity for Cyber Resiliency

  IBM Z Flexible Capacity for Cyber Resiliency is a new Capacity on Demand offering available on IBM z16™ servers. It enables you to shift capacity between participating IBM z16 machines at different sites and use the target configuration for up to one year. System Automation has been enhanced through OA64035 to support this new type of temporary capacity. You can use the ISQCCMD TCM command to add or remove temporary capacity for specific target hardware, and use the ISQCCMD TCDATA command to query the status and settings for a specific temporary capacity record.
- Activate LPAR with Load Parameters in ProcOps

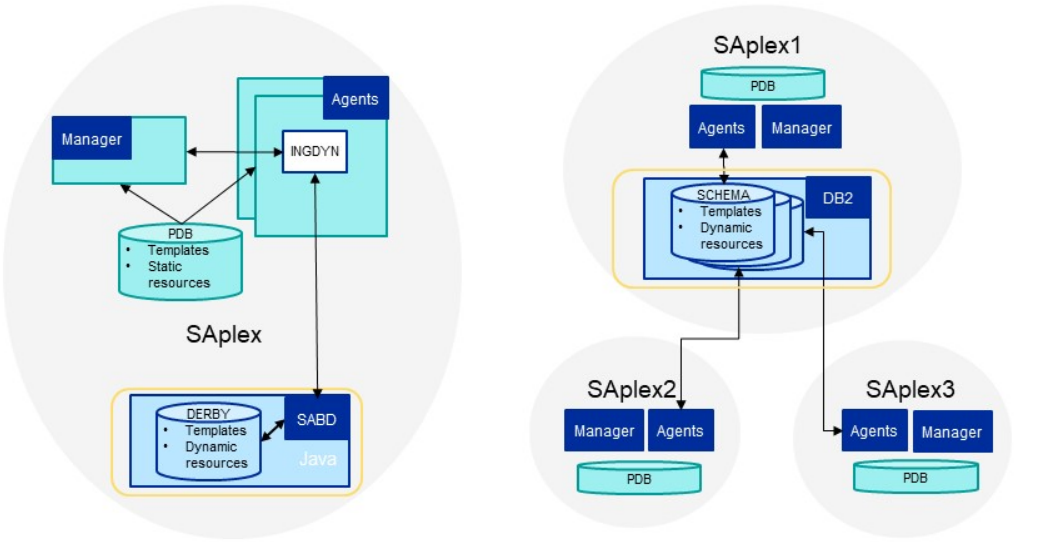
  This enhancement enables you to specify load parameters in the ACTIVATE command and perform both ACTIVATE and LOAD in one go.

OA63123 – Alternative Db2 data store of dynamic resources (Dec 2022)

The dynamic resources which are instantiated from templates by the INGDYN command are not part of the automation policy. They are previously stored and persisted in the System Automation data store only. This internal data store is a Java address space that is managed by System Automation, running once in every SAplex.

Now with OA63123, System Automation implements IBM Db2 for z/OS as an alternative option to store dynamic resources. It's only backend enhancement, without any changes to the INGDYN command. This Db2 backend option can bring you these benefits:

- High availability when using data sharing group.
- No need to manage additional (Java) address space(s) in the PDB.
- Improved INGDYN performance.
- Shifting processing away from NetView to Db2 engine by using stored procedures.



Database option 1: System Automation data store
Available with System Automation 4.3 GA version

Database option 2: IBM Db2 for z/OS
Available with System Automation 4.3 APAR OA63123

If you choose Db2 as the database, follow the configuration step "Configure Db2 as an alternative database of dynamic resources" in *Planning and Installation*.

Note: It doesn't allow you to switch back and forth between the database. It just allows you to choose between the two possibilities. Once you choose between any one, the data would be fetched and inserted into the database chosen.

Currently, there is no migration path from the internal data store to Db2. So if you choose Db2, you have to reconfigure and start anew.

Service Management Unite Enhancements (Dec 2022)

This information contains an overview of the major changes in IBM Service Management Unite Automation V1.1.9.2.

- Renamed the container command line utility `eezdocker` to `eezcontainer` to be independent of the underlying container runtime environment.
- Added the "Set Runmode" function. You can specify the name of the RunMode that you want to set, and the resources that are not qualified for the currently active RunMode will be stopped.

OA63831 – INGPLEX Enhancements (Nov 2022)

The following two enhancements are shipped with OA63831:

- Support new couple data set (CDS) types: LOGRY and LOGRZ
  - The INGPLEX CDS command now supports the new CDS types, LOGRY and LOGRZ, which are introduced in z/OS 2.4 and used specifically in GDPS KSYS environments. In addition, a **EHLQ** column is added on the **LOGR*x* Couple Data Set Information** panel, which is displayed via INGPLEX CDS TYPE=LOGR*x* (LOGR*x* refers to LOGR, LOGRY, or LOGRZ).
  - The SYSPLEX Policy Item panel is also changed, where you can specify additional LOGRY and LOGRZ data sets and alternative volumes. The policy is automatically converted when you're using a Customization Dialog at OA63831 service level.

    Note: Create a backup of your policy database. Once a policy is converted, it cannot be opened with a back-level dialog.
- The INGPLEX SYS command can display additional System Recovery Boost status in the new **Boost** column.

In addition, OA63831 introduces function level 3 to support new INGPLEX SYS and INGPLEX CDS columns. Make sure that all systems are at the current OA63831 service level before you set the function level to 3 in your NetView initialization member INGXINIT.

OA63538 – Enhanced inheritance of DB2 CONTROL policy item fields (Oct 2022)

With OA63538, inheritance rule of the DB2 CONTROL policy item is enhanced. Data in this policy item is now inherited per individual field, independent from each other. If you override any inherited values, those attributes become specific to that particular instance. Other attributes that are not overridden continue to inherit values from the linked class. This enhancement provides you more flexibility to define common attributes at class level, and override some attributes at instance level. For more details, see DB2 CONTROL Policy Item.

OA63014 – Policy Services Provider enhancements (July 2022)

With OA63014, the following enhancements are incorporated into the Policy Services Provider component.

- Two new parameters are added into the configuration file `ing.policyservices.environment` to support different character encoding.

```
ing.input.encoding=IBM-1047
ing.output.encoding=UTF-8
```

- Policy Services Provider's job log is optimized to eliminate redundant logging.

Service Management Unite V1.1.9.1 (June 2022)

- Improved the credential store handling to avoid that user IDs used to log in to the domain get revoked due to too many incorrect automatic login attempts after a credential change.
- Enhanced the Canzlog dashboard by supporting any date formats that are configured in IBM Z® NetView®, with APAR OA63249 installed in IBM Z System Automation.
- Enhanced the Restart and Request stop functions by showing affected resources (similar to what is shown by the INGREQ command) before you restart or stop a resource.

INGDYN enhancements (April-July 2022)

- With OA63309: Communication between INGDYN and the System Automation data store is enhanced to improve INGDYN performance.
- With OA63122: INGDYN CREATE now allows you to create a resource in the LINE mode only based on the latest template in the system. You can also use the same filters on the CREATE action as same as the LIST and INFO actions.
- With OA613019, a new RESOURCE parameter is added into the INGDYN command syntax, which you can use to list all generated resources for a certain template.

OA62554 – Processor Operations ISQSTOP enhancements (April 2022)

Previously, Processor Operations shutdown that was triggered by the ISQSTOP command might hang because some autotasks could not be stopped gracefully. For a proper cleanup of the Processor Operations environment, Processor Operations shutdown processing has been enhanced with OA62554 to identify hanging autotasks and force stop them if appropriate.

For these identified hanging autotasks, Processor Operations issues **STOP FORCE=*taskname*** commands, which might be issued twice, to force the autotasks to stop. If NetView indicates that the autotasks could be stopped in a normal manner, Processor Operations issues **STOP TASK=*taskname*** commands instead. The shutdown processing might be delayed up to 3 minutes. As a result of a **STOP FORCE** command, the corresponding autotask might abend with a system code X'EC4'.

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)

System Operations

System operations monitors and controls system operations applications and subsystems such as NetView, SDSF, JES, RMF, TSO, ACF/VTAM®, TCP/IP, CICS, Db2, IMS, ZWS, OMEGAMON and WebSphere®.

Enterprise monitoring is used by the SA z/OS Status Display Facility and through Automation Dashboards for Z Automation Web Console.

Processor Operations

Processor operations monitors and controls processor hardware and VM guest systems operations.

It provides a connection from a focal point system to a target processor Support Element or a Hardware Management Console. With NetView on the focal point system, processor operations automates operator and system consoles for monitoring and recovering target processors.

Processor operations performs or automates many operator tasks, usually done using the HMC, such as activate / deactivate a logical partition, power on and off, and reset of multiple target processors. You can initiate IPLs and respond to system startup operator prompt messages, monitor status, and detect and resolve wait states.

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 on page 20.

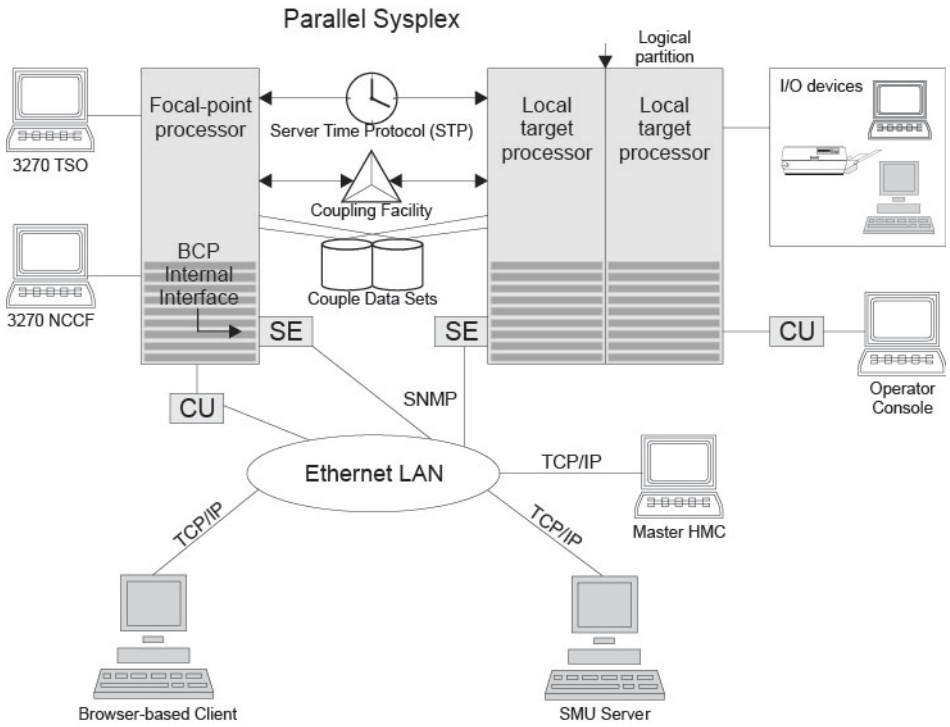


Figure 1. Basic Hardware Configuration

It shows a two processor Parallel Sysplex configuration, with systems running on it.

Operators can use a web browser to log on to IBM Z Automation Web Console to work with tabular and graphical views of the SA z/OS controlled resources. The IBM Z Automation Web Console dashboards receive status changes from any SA z/OS backend that is connected to Z Automation Web Console via the end-to-end adapter. Sysplex-specific facilities, like the coupling facility hardware can be managed and controlled using the 3270 Network Communications Control Facility (NCCF) based SA z/OS operator interfaces.

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 to perform hardware commands like LOAD or SYSTEM RESET to control the hardware and hardware images. SA z/OS processor operations can be configured to use TCP-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 sections describe some relevant resources that are used by SA z/OS and its components.

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.

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 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.

Server Time Protocol (STP)

Server Time Protocol (STP) is a server-wide facility that is implemented in the Licensed Internal Code (LIC) of the IBM Z processors. It provides time synchronization in sysplex or non-sysplex configuration.

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.

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 33.

SNMP

SNMP may be chosen as the hybrid protocol for communications between the processor operations focal point and the SE or HMC.

See also "Understanding the Processor Operations Hybrid SNMP Interface" on page 23.

BCP Internal Interface

For processor hardware automation in a sysplex environment, this link allows a z/OS system directly to communicate with its own hardware SE, as well as the SEs of other hardware 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 22.

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 24.

Looping Address Space Suppression

This is an automation solution ready for immediate use that queries IBM OMEGAMON (through its SOAP Interface) to detect address spaces that are in long running, CPU demanding execution patterns.

Such address spaces are probably caught in CPU intensive loops and are thus undesirable.

Once the procedure identifies such an address space, it will consult automation policy to categorize it and then it will apply the pass based recovery mechanism that is specified for the category. The recovery may be passive, diagnostic, active or an escalating mixture of two or three of those elements.

For further information, refer to “Planning for Looping Address Space Suppression” on page 41.

Planning the Hardware Interfaces

This section provides fundamental planning information about the required processor hardware consoles, considerations about processor hardware naming, and the hardware interfaces supported by Z System Automation.

Understanding the role of IBM Z hardware consoles for System Automation

The IBM Z hardware consoles provide web-based user interfaces, allowing operators and administrators to manually control IBM Z mainframes. For System Automation, both console devices Support Element (SE) and Hardware Management Console (HMC) are important. In addition to the hardware infrastructure and UI functions, they offer general-purpose mainframe operations management APIs for automation platform applications like System Automation. Hence, the consoles are key elements for availability and recovery turnaround time targets of IBM Z and its software stack.

Without access to a functioning SE or HMC for operations management tasks, System Automation commands and automation routines cannot monitor and control the hardware of a central processor complex (CPC) or any of its logical partitions (LPAR).

DISCLAIMER

System Automation cannot prevent human interventions from affecting console availability. With System Automation, you cannot automate, manage, or fully control the console device. The System Automation responsibility for the consoles is limited to the usage of the console application (HWMCA) SNMP APIs.

It is the sole responsibility of the IBM Z mainframe user or provider to make sure that the hardware consoles are available when needed by operations personnel, System Automation, or automation solutions based on System Automation services.

Related information

Appendix E, “Managing IBM Z console availability exceptions,” on page 219, provides additional information about how to handle console outages in System Automation environments.

Appendix F, “Planning to choose feasible CPC names,” on page 223, provides information about how the SE console name and the CPC name correlate, and gives recommendations about a useful naming scheme in System Automation environments.

Understanding the BCP Internal Interface

Z System Automation provides an IP network independent communication path, the Basic Control Program (BCP) internal interface, to manage and control processor hardware. This interface is available on IBM Z mainframes. You can use it to perform operations management commands like ACTIVATE, SYSRESET, and LOAD. You can also use it to monitor processor events like LPAR wait states or hardware messages for automation and alert forwarding purposes.

System Automation, when running in an LPAR of an IBM Z system, can target its own or other LPARs on the same processor, as well as other LPARs running on other processors connected to the same processor

LAN. The communication entry point is the Support Element (SE) of the IBM Z system. The LPAR that issues an operations management request is defined on the SE.

For the BCP internal interface (BCPii) communication, the IBM Z Hardware Management Console (HMC) acts as a request and response router between the IBM Z systems that are defined to it.

System Automationuses the BCPii communication with the following functions:

- Processor Operations (ProcOps)
- Sysplex Automation

In addition, BCPii internal services of System Automation are used by the IBM GTS service offering GDPS.

Understanding the Processor Operations Hybrid SNMP Interface

The IBM Z mainframe hardware family provides an operations management application programming interface, the IBM Z SNMP API. This interface uses a SNMP MIB based data model that is stored on the Support Elements or Hardware Management Consoles of the IBM Z mainframes. This API provides access to its functions and data by either an IP network, or the BCP internal interface (BCPii).

SNMP over IP

The IP network access allows you to address Support Elements or Hardware Management Consoles, using the available IP network infrastructure.

SNMP over BCPii

The BCPii access is bound to Support Elements only, without requiring any IP network infrastructure outside of IBM Z's own processor LAN.

At runtime, ProcOps can use only one access path to an IBM Z. For an inactive connection, its predefined access paths can be switched, which allows a hybrid connection operation mode. ProcOps itself remains up and running to service other active connections.

For IBM Z hardware operations management tasks like ACTIVATE, Temporary Capacity Change, SYSRESET, or LOAD, you can use the various ProcOps connection paths to fulfill different security demands for operations management automation.

SNMP Over IP: Understanding the Supported SNMP Versions

Simple Network Management Protocols (SNMP) is an "Internet-standard protocol for managing devices on IP networks". There are several versions of the protocol that are available. System Automation supports SNMPv2c and SNMPv3 protocols.

SNMPv2c is Community-based Simple Network Management Protocol version 2 and is defined in RFC 1901 - RFC 1908. Authentication is done based on a Community Name.

SNMPv3 makes no change to the protocol aside from the addition of cryptographic security and user and password authentication and provides more security compared to the SNMPv2c:

- Confidentiality - Encryption of packets to prevent snooping by an unauthorized source.
- Integrity - Message integrity to ensure that a packet has not been tampered with in transit including an optional packet replay protection mechanism.
- Authentication - to verify that the message is from a valid source.

Understanding the Hardware Console Automation Interface

The IBM Z mainframes provide a console facility that the SA z/OS hardware interfaces use to perform remotely either manual or automated operating system initialization and recovery.

See Appendix A, “Using the Hardware Integrated Console of IBM Z for External Automation with SA z/OS,” on page 189 for console definition, usage, performance, network, and basic information.

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

BCP internal interface (BCPii)

BCPii, which is defined as INTERNAL connection in System Automation Customization Dialog, is required in the following cases:

- You want to use the Parallel Sysplex enhancements of SA z/OS and you have configured your customization to use IXC102A message automation.
- You plan to use GDPS to monitor and control an IBM Z mainframe.

By design, the BCPii connections of the INTERNAL protocol have a peer-to-peer concept that tries to establish BCPii connections between all defined processors and systems in z/OS Sysplexes. You can use special processor settings in the Customization Dialog to limit the number of BCPii peer-to-peer connections that are automatically initialized.

ProcOps SNMP

If you do not use GDPS to monitor and control an IBM Z mainframe, you can configure the ProcOps SNMP connection to have its full operations management function set. Because of its hybrid connection capabilities, it can satisfy various network and security demands. ProcOps provides full support of the IBM Z emitted hardware events for automation and alert forwarding purposes. ProcOps can be activated on demand because it is implemented as a startable function within the System Automation product. By design, ProcOps acts as a focal point, requiring less active BCPii sessions than the peer-to-peer mode of the INTERNAL protocol does.

Using ProcOps SNMP and INTERNAL connections together

ProcOps SNMP and INTERNAL connection protocols can coexist. You can define both of them as processor connections. At run time, a single System Automation agent instance can have a ProcOps SNMP session (IP or hybrid over BCPii) in parallel to an INTERNAL (BCPii) session, targeting the same IBM Z processor.

From an automation perspective, it is important to understand that both sessions allow hardware automation. You need to make sure that the common active sessions do not cause automation conflicts. It is recommended to decide which IBM Z mainframe LPAR is monitored and controlled over ProcOps and which ones run, for example, under GDPS control. Choose common protocol operations for limited and tightly controlled use cases only.

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 13: Verify the Number of available REXX Environments” on page 103.

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

z/OS Considerations

Prefixes

Make sure that you do not have any load modules, REXX parts or members with the following prefixes.

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

Defining the XCF Group

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.

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" information in *IBM Z System Automation Defining Automation Policy*.

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 on page 26). 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.

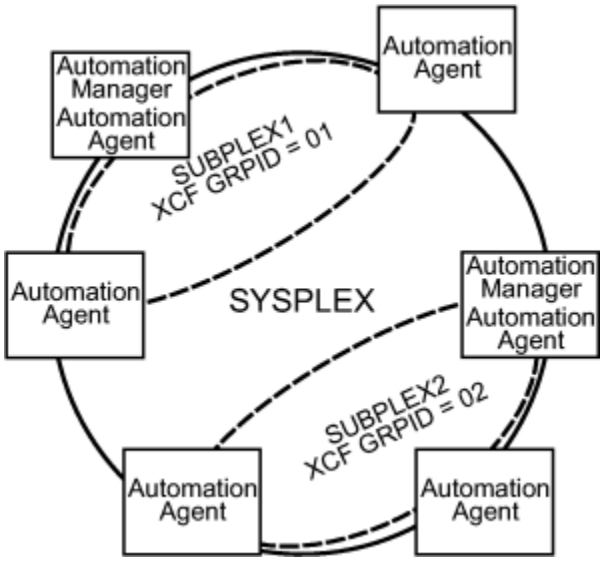


Figure 2. Using SA z/OS Subplexes

To allow automation agents within the same physical sysplex, but to communicate via XCF rather than NetView Gateways within different SA z/OS subplexes, optionally an extended XCF Communication Group can be defined as well. For more details, see “Defining Extended XCF Communication Groups” on page 26.

Defining Extended XCF Communication Groups

By default, an SA agent can only communicate with other agents that reside in the same SA subplex as described above. The introduction of the PLEXID parameter allows the extension of XCF communication between agents that reside in different SA subplexes.

The PLEXID parameter is a group suffix used to add the automation agent to an extended XCF communication group. This enables the automation agent to communicate via XCF with all other SA agents that were added to the same PLEXID group even though they are outside of the SA subplex. The PLEXID parameter may be defined in the member INGXINIT.

Along with the introduction of the PLEXID parameter, the TARGET parameter of all SA commands are enhanced such that they accept system name, domain id or SA subplex name of those agents that were added to the same PLEXID group. Use the command INGAMS in order to list all automation agents and automation managers that exist within the same PLEXID group.

If you want to use extended XCF communication it is strongly recommended to add all automation agents of the same SA subplex into the same PLEXID group.

If needed you may separate a specific SA subplex from the extended XCF communication group, see the figure in “Using SA z/OS Subplexes” on page 25. However, when you add all automation agents from all SA subplexes into the same PLEXID group then you have 'Single Point of Control' for all resources of all participating SA subplexes all over the physical Sysplex.

Another advantage of the enhanced XCF communication group is that you can reduce the number of gateway definitions. It helps to limit the number of NetView gateway definitions only to those SA agents that were really remote and not reachable via XCF, for example systems outside of the physical sysplex.

Message Delivery Considerations

SA z/OS relies on 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 commands 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 COMMNDxx PARMLIB member.

System Operations Considerations

Defining multiple SA z/OS instances on one system

When you define multiple SA z/OS instances on one system, follow the following rules to avoid impacting the serviceability of this product:

- Running multiple SA z/OS instances on one system where each of these instances automates SA z/OS resources is not supported.
- An SA z/OS instance for automation purposes is intended to run once on a system.

  If you run multiple instances of SA z/OS on one system, only one of those instances can perform automation tasks. This is important for GDPS environments where the automation tasks are performed by one SA z/OS instance (for example, within the GDPS Metro environment) while another instance serves as a base for GDPS only (for example, the GDPS Global – GM environment).
- Define gateway policies only for the SA z/OS instance that automates resources

  Gateways belong to the SA z/OS infrastructure. There is a one-to-one relationship between the SA z/OS automation agent NetView and the system that hosts the automation agent. They can be mapped against each other no matter you specify the system name or the NetView domain ID in a command's target parameter. This fact implies the limitation that only the SA z/OS instance responsible for automation tasks can be added to the SA z/OS gateway definitions. For the other instances, such as additional GDPS-related SA z/OS instances, you can not add their NetView domains in the gateway definitions.
- When running more than one SA z/OS instance, make sure that the underlying infrastructure is configured for the instance performing the automation tasks. This is required because certain parts of the infrastructure (MPF, MRT, PPI, LNK, LPA) exist only once per system.
- The MRT and PPIs reside within that SSI address space which is started first. Stopping this SSI will disable automation partly. Therefore, it is recommended to keep this SSI highly available.
- Any additional (NetView and SA z/OS) must match the releases residing in the LNKLST and LPA.

Not obeying these rules will prevent the ability to service this product.

SA z/OS initialization autotasks

SA z/OS ships two sample automation operators, AUTINIT1 and AUTINIT2. 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.

z/OS Communication Server and z/OS Language Environment (LE) considerations

SA z/OS relies on the setup of TCP/IP and the z/OS LE environment.

If CEE.SCEELKED is in LINKLIST or STEPLIB, TCPIP.SEZALOAD must be placed before CEE.SCEELKED. Failure to do so will result in a 0C1 system abend for the TCP/IP REXX socket calls.

SA z/OS Hardware Interface: Important Considerations

The SA z/OS processor support commands and modules of Processor Operations and the BCP Internal Interface require a NetView task environment of CMD LOW to operate.

If you plan to use CMD HIGH task environments, be aware that ProcOps and BCPii function commands will not operate in such task environments. The ProcOps or BCPii function command will end prematurely with an error message that identifies the cause of the problem.

However you can still use NetView tasks with a CMD HIGH set for other purposes.

Automation Manager Considerations

This information 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 Z System Automation 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 USS file system 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 109).

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 configured 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.
- Alternatively, add the parameter 'SUB=MSTR' to all ASNAME definitions that are not being moved to a separate member in the action listed above. An example for a definition update would be:

```
/*********************************************************/
/* 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. With z/OS version 1.10 or higher this restriction is solved and the Automation Manager can be initialized without JES and the need to be superuser. However BLOCKOMVS=YES still requires UID(0).

Recovery Concept for the Automation Manager

To ensure the automation manager 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).

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

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. It is important though that all automation managers (PAM and SAMs) run on systems which are in the same time zone.

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.

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 XCF for establishing communication between the automation manager and the automation agents, and a VSAM data set (the takeover file) for keeping a backup copy of the status of the automated resources.

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 30 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

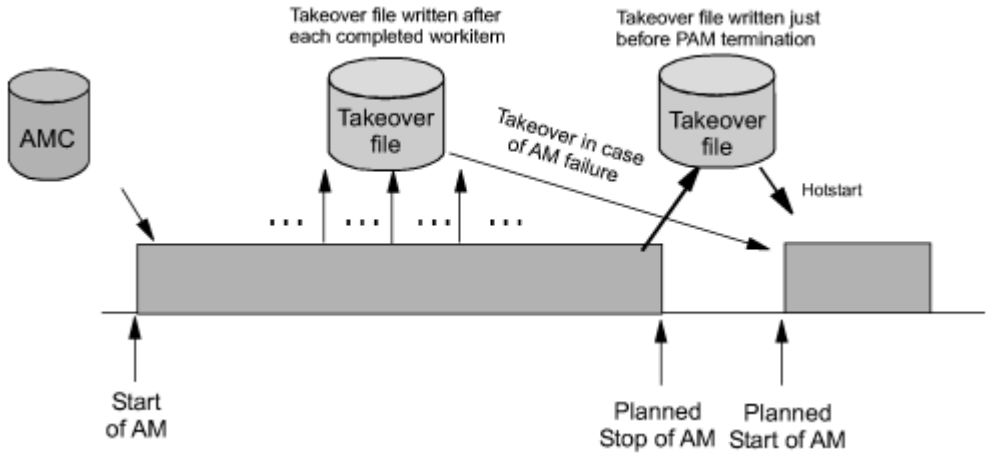


Figure 3. Using Only the Takeover File for Status Backup

Table 4 on page 30 outlines the various recovery scenarios.

Table 4. 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 hot start
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 hot start

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

This section contains information required for the installation of alert notification by SA z/OS.

Introduction of Alert Notification by SA z/OS

SA z/OS alert notification is triggered by the invocation of the INGALERT command.

It can be used to perform one ore more of the following tasks:

- Start notification escalation by IBM Tivoli System Automation for Integrated Operations Management (SA IOM)
- Display an event on a centralized operator console such as IBM Tivoli Enterprise Console (TEC) or IBM Tivoli Netcool/OMNIbus (OMNIbus)
- Create a trouble ticket in a service desk application such as IBM Tivoli Service Request Manager® (TSRM)
- Perform an arbitrary task in a user-defined alert handler

The following communication methods are available for alert notification:

- Use the peer-to-peer protocol of SA IOM to start a REXX script on the SA IOM server
- Send a Tivoli Event Integration Facility (EIF) event
- Send XML data the to IBM Tivoli Directory Integrator (TDI) and from there trigger the creation of the trouble ticket
- Pass parameters to the user-defined alert handler that is called as a NetView command

Note: EIF events and the TDI interface can be used to perform a variety of tasks or to integrate other operator consoles or service desk applications. The ones listed above are provided by SA z/OS as samples.

The behavior of INGALERT is controlled with the INGCNTL command at the system level, by a resource's Inform List at the resource level and even more granularly by CODE entries for the INGALERT entry in the MESSAGES/USER DATA policy item.

For details about the INGALERT and INGCNTL commands, see *IBM Z System Automation Programmer's Reference*.

Alert Notification Infrastructure in SA z/OS

When INGALERT is called in a SA z/OS subplex the system tries to reach all specified targets by passing the request from one agent to another.

If, for instance, INGALERT is called on SYS1 in order to start an SA IOM notification escalation, but SYS1 has no connection to the SA IOM server, the request is routed to SYS2 and SYS3 and so on, until the SA IOM server can be reached.

This implies that you need not have all the connectivity to your distributed products on each system in the subplex, although you should have it at least on one, of course. This is true for all of the communication methods mentioned in “Introduction of Alert Notification by SA z/OS” on page 31.

For details about the alert notification infrastructure see "Alert-Based Notification" in *IBM Z System Automation Customizing and Programming*.

Integration via SA IOM Peer-To-Peer Protocol

The integration of SA z/OS with SA IOM is based on the SA IOM peer-to-peer protocol.

This requires that the SA IOM server must accept the system running the SA z/OS agent (or agents) as valid peers. For details about setting up SA IOM, see *IBM Tivoli System Automation for Integrated Operations Management User's Guide*.

Through this protocol a REXX script is triggered on the SA IOM server that starts the notification escalation process asynchronously. A return code and eventually an error message are passed back to SA z/OS indicating whether the notification escalation could be started.

Note that it is not verified whether an operator can actually be notified by SA IOM.

To use integration via the SA IOM peer-to-peer protocol you must be able to set up a TCP/IP connection to the SA IOM server from at least one system that is running an SA z/OS agent.

See "Enabling Alert Notification via SA IOM Peer-To-Peer Protocol" on page 105.

Integration via EIF Events

SA z/OS can send out EIF events as the result of an INGALERT invocation. To create such an EIF event the message adapter or the confirmed message adapter of the IBM Tivoli Event/Automation Service (E/AS) is used via the program-to-program interface (PPI).

To use integration via EIF events there must be an E/AS on at least one system that is running an SA z/OS agent.

Because SA z/OS communicates only with E/AS it does not matter which product receives the EIF event and which platform it is running on. There is, however, some customization required for these products.

For more details about how to set up the E/AS and configure OMNIbus on Windows, see "Enabling Alert Notification via EIF Events" on page 105.

Integration via Trouble Ticket Information XML

When the creation of a trouble ticket is desired INGALERT sends XML data to a known URL (host and port). It is expected that the server sends back a response indicating success or failure and possibly an error message.

It is irrelevant what kind of server this is and which platform it runs on. However, it is recommended that the server is a TDI Runtime Server. Samples are provided for this server and the customization is described in "Enabling Alert Notification via XML" on page 107.

To use integration via trouble ticket XML you must be able to set up a TCP/IP connection to a TDI server from at least one system that is running an SA z/OS agent.

Integration by User-defined Alert Handler

When INGALERT is told to inform a user-defined alert handler it calls the specified command synchronously in the NetView environment.

Parameters are passed to the alert handler and a convention regarding return code and output messages must be obeyed. For details about the user-defined alert-handler, see INGALERT in *IBM Z System Automation Programmer's Reference*.

To use integration by user-defined alert handler, the code must be accessible from at least one system that is running an SA z/OS agent.

For more details see "Enabling Alert Notification via User-Defined Alert Handler" on page 107.

Chapter 5. Planning for Automation Connectivity

This information 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 and NetView facilities.

Defining System Operations Connectivity

This section discusses the following aspects of defining system operations connectivity:

- "Multiple NetViews" on page 33
- "Overview of Paths and Sessions" on page 33

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 33
- "Gateway Sessions" on page 34

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 "SDF Focal Point Monitoring" in *IBM Z System Automation 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 4 on page 34 shows a single gateway between two SA z/OS agents, ING01 and ING02.

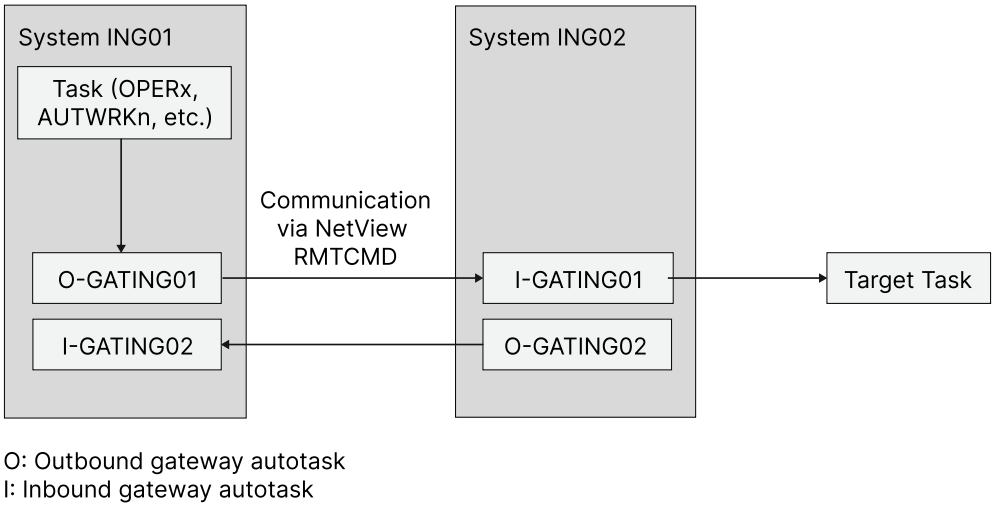


Figure 4. Single Gateway Example

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 ING01, the gateway task is GATING01.

When VTAM becomes active, the gateway task (GATOPER) issues a CONNECT call to the remote system, ING02 in our example. If the GATING01 task on the remote system is not already active, it will be started automatically by NetView.

All requests initiated by system ING01 and destined for system ING02 use the task pair GATING01. Likewise all requests that originate on system ING02 and are destined for system ING01 use the pair GATING02. 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 Z System Automation 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 Z System Automation 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 5 on page 35 shows three systems: CHI01, ATL01, and ATL02. System CHI01 is the focal point for forwarding messages from target systems ATL01 and ATL02. In Figure 5 on page 35, gateways are designated as follows:

O
Outbound gateway autotask

I
Inbound gateway autotask.

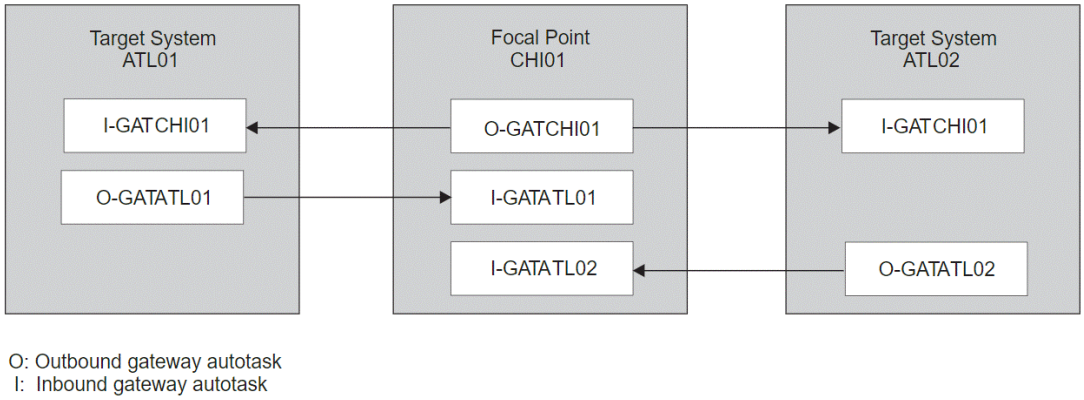


Figure 5. Example Gateways

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 that is executed periodically. The time interval is set in the **Gateway Monitor Time** field in the SYSTEM INFO policy item for the System policy object.

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.

The "FULL SESSIONS Policy Item" topic in *IBM Z System Automation 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.

For details on these operator activities, see "Communicating with Other Systems" in *IBM Z System Automation 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:

- 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

IBM Z processors have a second Support Element (SE) 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 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 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 configured 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

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

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

Message monitor tasks receive SNMP traps from the Support Element's SNMP clients, 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, Polling and General Management 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.

The general management task is used for message automation in case the recovery task is not available because of other workloads.

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” on page 38 and “Preparing the Alternate Focal Point System Connections” on page 39 for focal point system connections
- “Preparing the Target System Connections” on page 40 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:

- 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 System z® 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. In addition, the well-known port for SNMP, port 161, must be configured to allow TCP packet traffic between ProcOps and the SE/HMC.

If your firewall has session keep alive rules activated to control inactive sessions, you can define a keep alive idle period for a ProcOps event connection. Once an idle period expires, the connection end point (SE/HMC) automatically issues a ProcOps keep alive event and sends it to the waiting z System API on the

Procops FP system. If the ProcOps defined idle time period is shorter than the time defined in the firewall rule, the event session remains active. The default is no defined ProcOps idle time. The required minimum SE/HMC code level for the idle time support is console version 2.13.0, available with IBM z13.

ProcOps connection idle times are defined using Advance Automation CGLOBAL variable AOF_AAO_ISQ_KALIST. For more information, refer to the AOF_AAO_ISQ_KALIST variable in the table "Global Variables to Enable Advanced Automation (CGLOBALS)" in the appendix "Read/Write Variables" of *IBM Z System Automation Customizing and Programming*.

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:

- SNMP
- TCP/IP

Connection Example

Figure 6 on page 39 shows an alternate focal point system as well as a primary focal point system connected from an IP network to the processor hardware LAN.

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.

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

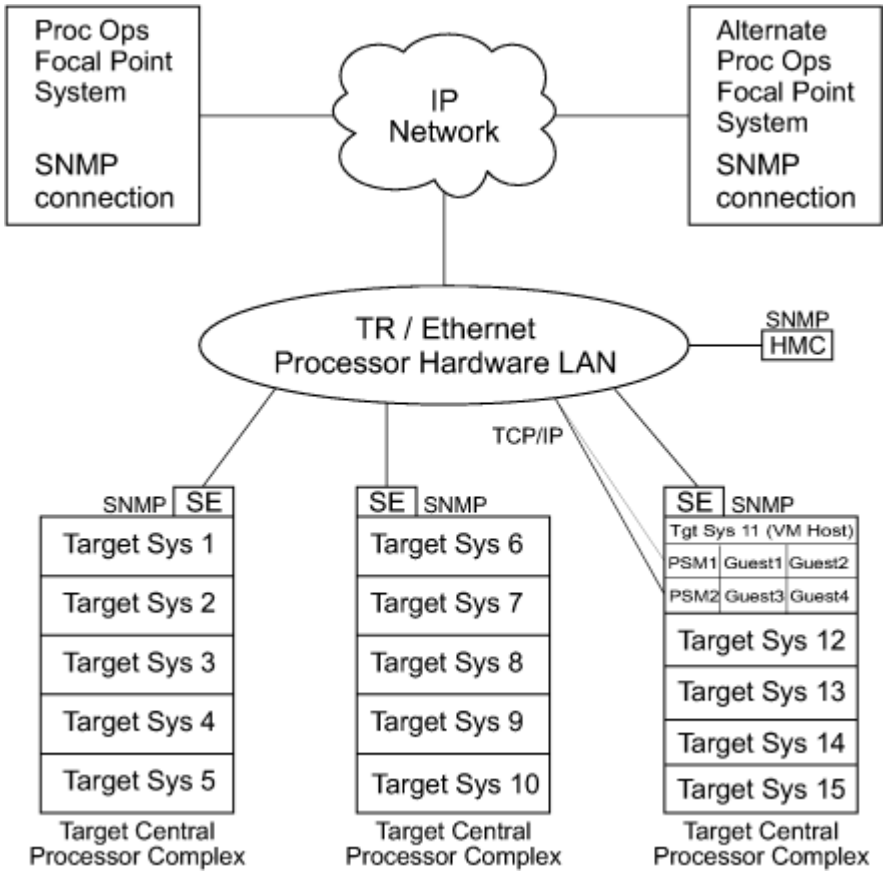


Figure 6. Alternate and Primary Focal Point System Connections from an IP Network to the Processor Hardware LAN

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 management interfaces of the SE or HMC are used to perform tasks like SYSTEM RESET, LOAD, TEMP.CAPACITY or ACTIVATE.

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.

Chapter 6. Planning for Integration with IBM Tivoli Monitoring

Planning for SOAP over HTTPS

The default communication for talking over the SOAP interface is by a HTTP link. As this is not secure, there is also the option of setting up an HTTPS connection.

To do this you will need to modify the parameters for your TCPIP stack.

For further information, refer to “Step 38A: Enabling SOAP over HTTPS for a TEMS” on page 148.

Planning for Looping Address Space Suppression

Prerequisites

In order to use the Looping Address Space Suppression automation you need:

- IBM OMEGAMON for z/OS installed and running
- A Tivoli Enterprise Monitoring Server (TEMS) that is receiving data from the IBM OMEGAMON for z/OS instance and which is running a SOAP server
- If you wish to use HTTPS communication between SA z/OS and the Tivoli Enterprise Monitoring Server, you need to plan for it to be enabled.

Data Gathering

Ask your application programmers and batch schedulers if they have any batch jobs or other programs that run for a long time in CPU intensive loops. If they can identify them now, you can enter them into your automation policy at the definition stage as known false positives which should avoid accidents later on.

Even though you should start running with automation set to LOG only, it is better to obtain a list of predicted false positives from your application experts than to simply hope that they all appear during your test phase.

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

Activation profiles (Reset, Image, Load) names are processor (CPC) related and are manually defined at the HMC or SE.

When doing this with processor operations, note that these names must consist of the characters A-Z and 0-9. Image (LPAR) profile names can be up to eight characters long; Reset and Load profiles can have a length of up to sixteen characters.

Part 2. Installation and Configuration

This part provides instructions for:

Chapter 8. SMP/E Installation

About this task

SysOps	ProcOps
√	√

1. Perform the SMP/E installation on the appropriate system. Apply any available maintenance.
2. The security administrator ensures that the system programmer has ALTER access to the HLQs where they are to deploy the SMP/E target libraries to. You must use a system where the Customization Dialog is run and the systems where SA z/OS is deployed for automation.
3. If the SA z/OS installation is planned on other systems than the SMP/E system, then the system programmer must transmit the SMP/E target libraries to the system where the Customization Dialog is run and the systems where SA z/OS is deployed for automation. You can duplicate these data sets or wait until you must distribute an update to them. Updating the data sets that the product is using is not recommended.
4. On the system where the Customization Dialog is run, the system programmer must make the INGEDLG routine available to the automation administrator under ISPF. It is suggested that access to the Customization Dialog is restricted as the automation policies that are used to edit, compose part of your operation runtime data.
5. The security administrator must provide the following permissions for the automation administrator:
   - READ access to the SMP/E target libraries
   - ALTER access to the HLQ used for the Automation Policy Databases (PDB) and the automation control data sets.

Table 5 on page 47 shows a list of target data sets as provided by the SMP/E installation process to be used for production on your system.

Table 5. Target Data Sets

Data Set Name	Description
ING.SINGINST	SMP/E jobs to install the product alternatively to using SMP/E dialogs 1
ING.SINGLINK	Different SA z/OS modules in LINKLST 2
ING.SINGLOAD	Different SA z/OS modules 2
ING.SINGLODE	Different SA z/OS modules 2 For Policy Services Provider support 7
ING.SINGLPA	Different SA z/OS modules in LPALIB 2
ING.SINGMENU	ISPF messages 3 NetView messages 2
ING.SINGMENV	For VM second level systems support 4
ING.SINGMJPN	Kanji NetView messages 5
ING.SINGOBJV	For VM second level systems support 4
ING.SINGPARM	NetView DSIPARM samples 2
ING.SINGPDB	Policy database samples 3

Table 5. Target Data Sets (continued)

Data Set Name	Description
ING.SINGPENU	ISPF panels 3 NetView panels 2
ING.SINGPJPN	Kanji NetView panels 5
ING.SINGPRF	NetView profiles 2
ING.SINGREXX	NetView REXX execs 2
ING.SINGREXV	For VM second level systems support 4
ING.SINGSAMP	General samples 2
ING.SINGSENU	ISPF skeletons 3
ING.SINGTENU	ISPF tables 3
ING.SINGTREX	TSO REXX execs 6 ISPF REXX execs 3

Table 6 on page 48 shows a list of the USS directories that are provided by the SMP/E installation process.

Table 6. USS Paths

USS Path	Description
/usr/lpp/ing/adapter	Shell script 8
/usr/lpp/ing/adapter/lib	Executable 8
/usr/lpp/ing/adapter/config	Configuration files 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/datastore	For the System Automation data store that is required for dynamic resources 9
/usr/lpp/ing/datastore/lib	Executable 9
/usr/lpp/ing/datastore/config	Configuration files 9
/usr/lpp/ing/dist	For distributed connectors 10
/usr/lpp/ing/dist/tec	Tivoli Enterprise Console (TEC) related code 10
/usr/lpp/ing/dist/tdi	Tivoli Directory Integrator (TDI) related code 10
/usr/lpp/ing/dist/omnibus	Tivoli Netcool/OMNIbus-related code 10
/usr/lpp/ing/dist/ZOWE	For the System Automation Plug-in for Zowe CLI 12
/usr/lpp/ing/infobroker	For the System Automation Information Broker 13
/usr/lpp/ing/infobroker/bin	Empty 13
/usr/lpp/ing/infobroker/lib	Executable 13
/usr/lpp/ing/infobroker/config	Sample configuration files 13

Table 6. USS Paths (continued)

USS Path	Description
/usr/lpp/ing/policyservices	For the System Automation Policy Services Provider 14
/usr/lpp/ing/policyservices/lib	Policy Services Provider application's jar file 14
/usr/lpp/ing/policyservices/config	Sample configuration file 14
/usr/lpp/ing/restsrvr	For the System Automation Operations REST Server 11
/usr/lpp/ing/restsrvr/bin	Empty 11
/usr/lpp/ing/restsrvr/config	Sample configuration files 11
/usr/lpp/ing/restsrvr/lib	Executable and deployable sources 11
/usr/lpp/ing/restsrvr/samples	Miscellaneous samples to use or integrate with System Automation Operations REST Server 11
/usr/lpp/ing/sap	SAP-related code 10
/usr/lpp/ing/ussauto	Customer data/empty at installation 8
/usr/lpp/ing/ussauto/lib	USS automation executable file 8

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

1
Data sets of this category need to be accessed by the system programmer running SMP/E.

2
Data sets of this category need to be used by the NetView and automation team responsible for setting up and customizing system automation.

3
Data sets of this category are related to ISPF and need to be accessed by everyone that uses the customization dialog.

4
Data sets of this category are defined in VM setup.

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

6
Data sets of this category need to be accessed by everyone that uses the SA TSO REXX environment.

7
This is a PDSE data set that contains program objects that are currently used only for the Policy Services Provider. It must not be APF-authorized.

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

9
Files in these directories are required for the System Automation data store, which is a prerequisite of dynamic resources.

10
Files in these directories are used to integrate with other products.

11
Files in these directories are used for the System Automation Operations REST Server.

12
Files in these directories are required for the System Automation Plug-in for Zowe CLI.

13

Files in these directories are required for the System Automation Information Broker.

14

Files in these directories are required for the System Automation Policy Services Provider.

Chapter 9. Base SA z/OS Configuration Using the Configuration Assistant

The configuration of this product is supported by the Configuration Assistant.

Instead of manually adapting configuration jobs, start procedures, and initialization files to your environment, this assistant generates these files for you. The settings that are implemented are taken from the user-customized `INGDOPT Configuration Options` file.

The generated files are created as members within a dynamically allocated configuration data set (CONFLIB). In this data set, they are populated with the values that you define in the `INGDOPT Configuration Options` file.

The `CONFLIB` data set contains these items:

- Jobs to allocate all data sets and USS paths that are required by SA z/OS during runtime
- Procedures to start the components of SA z/OS to be copied to your target `SYS1.PROCLIB`
- Runtime configuration members for both Automation Manager and Automation Agent
- Parameter files that are ready to be copied to your target `SYS1.PARMLIB`
- VTAM definitions that are files ready to be copied to your target `VTAMLST`
- Jobs to delete data set files and USS paths in case you must reconfigure or delete SA z/OS again
- A job to verify the success of the installation and configuration process.

All members within the `CONFLIB` data set can be inspected, if required. If you applied changes to the generated members, be aware that the `CONFLIB` data set is newly allocated when running the configuration assistant another time.

Note: The security administrator must give the system programmer and the automation administrator ALTER access to the HLQ for the locally allocated and active automation policy data sets. The security administrator must also authorize the user ID used by the SA z/OS started tasks for accessing the data sets as follows:

- READ for SMP/E and active automation policy
- UPDATE for the locally allocated data sets

Table 7 on page 51 serves as a reference to manual SA z/OS configuration steps as documented below in Chapter 10, “Traditional SA z/OS Configuration,” on page 67. All steps are marked which are covered by the configuration assistant.

Table 7. SA z/OS Host Configuration Tasks supported by the Configuration Assistant . ✓ = supported

Task	SysOps	ProcOps
Step 1: SMP/E Installation. Refer to Chapter 8, “SMP/E Installation,” on page 47.		
“Step 2: Allocate System-Unique Data Sets” on page 69	✓	✓
“Step 3: Allocate Data Sets for the ISPF Dialog” on page 72	✓	✓
“Step 4: Configure SYS1.PARMLIB Members” on page 73	✓	✓
“Step 5: Configure SYS1.PROCLIB Members” on page 76	✓	✓
“Step 6: Configure NetView” on page 78	✓	✓
“Step 7: Preparing the Hardware” on page 84		
“Step 8: Preparing the VM PSM” on page 92		

Table 7. SA z/OS Host Configuration Tasks supported by the Configuration Assistant . ✓ = supported (continued)

Task	SysOps	ProcOps
"Step 9: Configure the Automation Manager" on page 95	✓	n/a
"Step 10: Configure the Component Trace" on page 97	✓	
"Step 11: Configure the System Logger" on page 97		
"Step 12: Configure ISPF Dialog Panels" on page 98	✓	✓
"Step 13: Verify the Number of available REXX Environments" on page 103		
"Step 14: Configure Function Packages for TSO" on page 103		
"Step 15: Configure Alert Notification for SA z/OS" on page 104	✓[1]	
"Step 16: Compile SA z/OS REXX Procedures" on page 108		
"Step 17: Defining Automation Policy" on page 108		
"Step 18: Define Host-to-Host Communications" on page 109	✓	✓
"Step 19: Enabling SA z/OS to Restart Automatic Restart Manager Enabled Subsystems" on page 110		
"Step 20: Define Security" on page 111	✓	
"Step 21: Configure the Status Display Facility (SDF)" on page 111	✓[2]	
"Step 23: Check for Required IPL" on page 117	✓	✓
"Step 24: Automate System Operations Startup" on page 118	✓	✓
"Step 25: Verify Automatic System Operations Startup" on page 119		
"Step 26: Configure USS Automation" on page 120		
"Step 31: Enable the End-to-End Automation and Connect an SAplex to IBM Z Automation Web Console" on page 140	✓	
....		

1. Alert Notification through the Tivoli Event Integration Facility (EIF)
2. SDF configured for the local system

Preparing to Configure System Automation

Preparation consists of the following steps:

1. Allocate a data set where you can maintain working copies of the `INGDOPT Configuration Options` file and the Configuration Assistant job. See "Allocate a data set for work files" on page 53.
2. Create a work copy of the `INGDOPT Configuration Options` file and the Configuration Assistant sample job (INGDCONF). See "Create Work Copies" on page 53.
3. Edit the working copy of the `INGDOPT Configuration Options` file to reflect the parameters of the installed environment. These parameters are then used to build the necessary artifacts to complete

the configuration. See “Editing the Work Copy of the INGDOPT Configuration Options File ” on page 53.

4. Edit and submit the work copy of the INGDCONF sample job. This job allocates the CONFLIB data set and configures the rest of the configuration jobs. See “Editing and Submitting the Work Copy of the INGDCONF Configuration Assistant Job” on page 54.
5. Follow the instructions documented in CONFLIB in $INGREAD.

Note: The user ID under which these jobs are submitted must be authorized to read the SMP/E target libraries. Runtime-specific data sets are allocated with a high-level qualifier as is specified in the INGDOPT Configuration Options file. The user must have ALTER access to create these data sets.

Allocate a data set for work files

Allocate a data set where you can maintain working copies of the INGDOPT Configuration Options file and the Configuration Assistant job.

Compose the name of that library out of a high-level qualifier (HLQ), the SAplex name (SAplex) both of your choice and the low-level qualifier (LLQ) named CONFWRK. You cannot change this naming scheme because it is used by the Configuration Assistant job. For example, if you decide to use the HLQ of 'USER' and you configure SA z/OS on z/OS systems belonging to a sysplex named SYSPLEX1, the recommended name for the work data set is USER.SYSPLEX1.CONFWRK.

The length of the data set name cannot exceed 35 characters because the following data sets are allocated by other JCLs later on:

```
hlq.saplex.CONFLIB.&SYSNAME.
hlq.saplex.CONFLIB.VTAMLIB
```

&SYSNAME. represents a system symbol which is resolved when running these JCLs on the individual systems.

The characteristics for the data set (PDS or PDSE) are as follows:

```
RECFM=FB,LRECL=80
```

As an initial size for the CONFWRK data set, you might allocate the following number of tracks:

```
Primary Quantity . . 15
Secondary Quantity . 5
Directory Blocks . . 5
Block Size . . . . . 27920
```

Create Work Copies

The INGDOPT Configuration Options file and the INGDCONF Configuration Assistant are supplied as members in the sample data set that is part of SMP/E DDDEF name SINGSAMP.

Create a work copy of the INGDOPT and INGDCONF members in the work data set, which you allocated in the previous step. If you plan to configure more than one system, it's recommended to use system symbols in the INGDOPT copy. In this case, you need only one INGDOPT copy, which is processed by one INGDCONF JCL, for all the z/OS systems in an SAplex.

Do not change the members in the data set that belongs to SMP/E DDDEF SINGSAMP.

Editing the Work Copy of the INGDOPT Configuration Options File

You define various settings that vary from installation to installation in the INGDOPT Configuration Options file. Typical examples are data set high-level qualifiers, system name, and the NetView domain name. These settings are used to build the configuration files in the CONFLIB data set.

Next, edit the INGDOPT Configuration Options file according to the syntax rules and the documentation that you find within that file.

The INGDOPT Configuration Options file contains comprehensive documentation on the purpose of the parameters.

Editing and Submitting the Work Copy of the INGDCONF Configuration Assistant Job

This job runs the Configuration Assistant and allocates the CONFLIB partitioned data set.

The data set stores the generated JCLs, start procedures, parmlib members, and other initialization and configuration members. Follow the instructions that are given in the INGDCONF job to adapt the job statements and the JCL variables within your INGDCONF work copy. When finished, submit the job.

Follow the Instructions as Documented in $INGREAD

Documentation member $INGREAD was tailored to your installation and created in the CONFLIB data set.

Follow the instructions documented there and complete the basic configuration. When you are finished with $INGREAD, proceed with the configuration described in these sections.

Completing Member Configuration

Configure the System Logger (optional)

Configuring the System Logger allows gathering resource-related history data. Even though this configuration is not mandatory for resource automation, it is recommended for problem determination tasks.

This step must be performed on the target system, where SA z/OS is to be configured. See the configuration step “Step 11: Configure the System Logger” on page 97 in "Traditional SA z/OS Configuration" in *IBM Z System Automation Planning and Installation*.

Note: If the system logger is not configured, the INGDVRFY verification job issues a warning message. Ignore that message if you do not want to configure the system logger for automation.

Update SMFPRMxx (optional)

If you plan to use SMF records for the availability reporting of automated resources, you must update the SMFPRMxx member.

This step must be performed on the target system, where SA z/OS is to be configured. See the configuration step “Step 4I: Update SMFPRMxx” on page 76" in the "Traditional SA z/OS Configuration" section.

Install the TSO REXX Function Package (optional)

The function package is used for the following functions:

- Batch interface (see also member EVJSJ001 in *.SINGSAMP library)
- Relational Data Services (RDS)
- Syntax checking for automation table overrides

If you plan to use these functions, you must configure the TSO REXX Function Package on the target system where SA z/OS is to be installed.

See the configuration step “Step 14: Configure Function Packages for TSO” on page 103 in Chapter 10, “Traditional SA z/OS Configuration,” on page 67.

Configuration of Alert Notification for SA z/OS (optional)

SA z/OS provides an alert-based notification service that alerts subject matter experts. You can escalate automation problems that require manual intervention by sending alerts, events, or trouble tickets to different kinds of notification targets.

For more information, see "Alert-Based Notification" in *IBM Z System Automation Customizing and Programming*.

Automation Dashboards for Z Automation Web Console (optional)

Automation Dashboards for Z Automation Web Console is an optional customizable service management user interface that provides dashboards to operate IBM Z environments. Operators can quickly and confidently analyze, isolate, and diagnose problems. This user interface also enables operators to interact directly with systems that may be located in different SAplexes and even non-IBM-Z systems (using the Universal Automation Adapter), without going to a different console.

Automation Dashboards for Z Automation Web Console communicates with z/OS systems, that are managed by IBM Z System Automation, through an adapter, which is commonly called 'E2E adapter'.

For more information, refer to:

- *IBM Z System Automation End-to-End Automation*
- *Service Management Unite Installation and Configuration Guide*

End-to-End Automation (optional)

System Automation provides cross sysplex and cross platform automation capabilities. It allows automating resources across different SAplexes or across different platforms. For more information, refer to *IBM Z System Automation End-to-End Automation*.

Verifying Your Configuration

Submit the INGDVRFY Configuration Verification job on the target system where SA z/OS was configured.

This job is in the CONFLIB library. After the job terminates, investigate the job log for INGVxxxx messages. If required, correct the configuration according to those messages.

Start SA z/OS for the first time

Before you proceed to details about the contents of the automation policy and techniques in the Customization Dialog for resource definitions, use this section to get a jump-start with a correct Policy Database (PDB) for a plain z/OS system.

You can use the procedure to complete the initial configuration as explained previously. This procedure is expected to take less than 30 minutes.

After you validate your configuration and you have a basic policy, then you can skip the section.

Quick planning exercise

The created basic policy contains a number of standard applications (started tasks) on z/OS systems. The started tasks must match the naming standards that are in place on the target system.

The following planning sheet guides you to identify the real job names that are used in the PDB and ensures that the applications are named correctly.

Table 8. Worksheet for job names

Application	Description	Default Job Name	Real Job Name	Default Procedure Name	Real Procedure Name
AM	Automation Manager	AM		INGEAMSA	See Note 1
AM2	Spare Automation Manager	AM2		INGEAMSA	See Note 2
APPC	Advanced Peer-to-Peer Communication	APPC			
ASCH	APPC Scheduler	ASCH			
BLSJPRMI	Build SNAP Tables for IPCS	BLSJPRMI			
DLF	Data Lookaside Facility	DLF			
FFST	First Failure Support Technology	FFST			
HSM	Hierarchical Storage Manager	HSM			
IRRDPTAB	RACF dynamic parse table loader	IRRDPTAB			
JES2	Job Entry Subsystem 2	JES			
LLA	Library Lookaside	LLA			
OAM	Object Access Method	OAM			
OMPROUTE	Open MVS MultiProtocol Routing Daemon	OMPROUTE			
OMVS	UNIX System Services subsystem	OMVS			
RACF	Resource Access Control Facility	RACF			
RESOLVER	TCP/IP Name Resolver	RESOLVER			
RMF	Resource Measurement Facility	RMF			
RMFGAT	RMF Monitor III Data Gatherer	RMFGAT			

Table 8. Worksheet for job names (continued)

Application	Description	Default Job Name	Real Job Name	Default Procedure Name	Real Procedure Name
RRS	Resource Recovery Services	RRS			
SYSVAPPL	Automation Application	&JOBNAME		INGENVSA	See Note 3
SYSVIPLC	IPL Data Gatherer	SYSVIPLC		HSAPIPLC	See Note 4
SYSVSSI	Automation Subsystem Interface	SYSVSSI			
TCPIP	TCP/IP	TCPIP			
TSO	Time Sharing Option	TSO			
VLF	Virtual Lookaside Facility	VLF			
VTAM	Virtual Telecommun-ication Access Method	VTAM			
ZFS	z/OS File System	ZFS			

Notes:

1. When you specified `sa_am_start_proc` in the Options File, use this value, otherwise use what is specified for `sa_am_start_job.1`
2. When you specified `sa_am_start_proc` in the Options File, use this value, otherwise use what is specified for `sa_am_start_job.2`
3. When you specified `sa_saagent_start_proc` in the Options File, use this value, otherwise use what is specified for `sa_saagent_start_job`
4. When you specified `sa_ipldata_start_proc` in the Options File, use this value, otherwise use what is specified for `sa_ipldata_start_job`

In all likelihood, most of the listed applications are not changed because most installations already use the default names. However, for some applications, different job names might be used and therefore the job name attribute for such applications has to be adopted in the basic policy. Also, some of the applications might not exist on the target system, so those applications can be deleted or unlinked from the basic policy. Take note of those applications that require a job name change or that can be deleted.

Starting the Customization Dialog

The Configuration Assistant provided you with a REXX script called `INGEDLG`.

Procedure

1. Copy this script into a data set in your SYSPROC or SYSEXEC concatenation of your TSO session.
2. Start it as follows:

- %INGEDLG
- Alternatively, start it directly out of the CONFLIB with the TSO EXEC command. For example: TSO EXEC 'MYHLQ.SYSA.CONFLIB(INGEDLG)' After INGEDLG is started, you see a panel as follows:

```
  MENU  OPTIONS  HELP
--------------------------------------------------------------------------------
            IBM Z System Automation 4.4 Customization Dialog
Option ===> ____________________________________________________________

  0  Settings        User parameters

  BR Browse          Browse the Policy Database
  1  Edit            Edit the Policy Database
  2  Build           Build functions for Policy Database
  3  Report          Generate reports from Policy Database
  4  Policies        Maintain Policy Database list
  5  Data Management Import policies into a Policy Database
  U  User            User-defined selections

  X  Exit            Terminate Customization Dialog

  To switch to another Policy Database, specify the Policy Database name
  in the following field, or specify a ? to get a selection list.
  Current Policy Database . . . ____________________
                    Licensed Materials - Property of IBM
```

Creating a basic PDB

Firstly, you need to create a policy database (PDB).

Procedure

1. From the **IBM Z System Automation 4.4 Customization Dialog** panel, enter ? in the **Current Policy Database** field at the bottom of the page and press Enter.

   You see a panel as follows:

```
MENU  COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
                            Policy Database Selection             Row 1 of 23
Command ===> _____________________________________________  SCROLL===> PAGE

Action     Policy Database      Enterprise Name
****************************** Bottom of data ********************************
PF 1=HELP      2=SPLIT     3=END       4=RETURN    5=RFIND     6=RCHANGE
PF 7=UP        8=DOWN      9=SWAP     10=LEFT     11=RIGHT    12=RETRIEVE
```

2. To create a PDB, type the word new on the command line and press Enter.

   You now see a panel as follows:

```
  COMMANDS   ACTIONS   HELP
------------------------------------------------------------------------------
                          Create a New Policy Database             Row 1 of 1
Command ===> _____________________________________________________________

To define a new Policy Database, specify the following information:
   Policy Database Name . . _________________
   Enterprise Name. . . . . _________________
   Data Set Name. . . . . . _____________________________________
   Description  . . . . . . _____________________________________

   Model Policy Database. . *EMPTY_____________ Policy Database name or "?"
                                                for list of names
Add-on policies to be added to a standard SA model policy database:
Action     Status      Add-on Policy         Customizable
________               *BASE                 YES
________               *CICS
________               *DB2
________               *E2E                  YES
________               *GDPS
________               *HYPERSWAP
________               *IBMCOMP              YES
________               *IMS
________               *ITM                  YES
________               *PROCOPS
________               *SAPSRV
________               *TBSM
________               *ZWS

****************************** Bottom of data ********************************


PF 1=HELP      2=SPLIT     3=END       4=RETURN    5=RFIND     6=RCHANGE
PF 7=UP        8=DOWN      9=SWAP     10=LEFT     11=RIGHT    12=RETRIEVE
```

3. In the **Policy Database Name** field, enter the name of the PDB. This value must be a single word but can include underscores. TEST_PDB is recommended.
4. In the **Enterprise Name** field, enter the name of your business or the section of it that you are going to define in the PDB. This value must be a single word but can include underscores. TEST_SYSTEMS is recommended.
5. In the **Data Set Name** field, enter the name of the data set on disk that holds the policy database. A useful convention is to have the name end with a .PDB extension, and to use the same name with a .SOCNTL extension for the Automation Control File that gets built from it. If you enter a value without single quotation marks, it is taken to be relative to your TSO user ID. If you enter a value with single quotation marks, it is taken as an absolute fully qualified data set name. For example, TEST.PDB might result in data set *USER*.TEST.PDB, while `'AUTO.TEST.PDB'` results in a data set `'AUTO.TEST.PDB'`.

   Use what you specified for `sa_automation_policy` in the `INGDOPT Configuration Assistant Options` file and put single quotation marks around it. The section at the bottom with the add-on policies adds the sample policies to your empty policy database.
6. In the **Descripion** field, specify a comment for the PDB, for example the purpose of this PDB, to help you easily distinguish it from other PDBs.
7. Enter C in front of *BASE and press Enter.

```
                    Select Add-on Policy Components         Row 1 to 13 of 13
Command ===>  _____________________________________________________SCROLL===> CSR

Components of Add-on Policy : *BASE

Select one or more components to be added to your Policy Database:

Action Status      Component
_______ SELECTED    Base z/OS
_______ SELECTED    Job Entry Subsystem 2 (JES2)
_______ SELECTED    Job Entry Subsystem 3 (JES3)
****************************** Bottom of data ********************************
```

For the basic PDB, only the *Base z/OS* components and one of the JES subsystems are required.

8. To deselect the component, which is not required, specify M in front and press Enter.

   The **SELECTED** status is now only shown for *Base z/OS* and either for JES2 or for JES3. It depends on what type of JES that you use on the target system.
9. When finished, press PF3.
10. Press Enter to review the contents of the **New Policy Database Dataset Information** panel and press Enter once more to create the policy.

    After a few messages (press Enter to clear them), you find yourself on the **Entry Type Selection** panel for your new policy database:

```
                              Entry Type Selection
Option ===>  ____________________________________________________________________

Enter number or entry type or use "BR <entry type>" for browse

  1 ENT    Enterprise                     30 TMR    Timers
  2 GRP    Groups                         32 TPA    Tape Attendance
  3 SBG    SubGroups                      33 MVC    MVS Components
  4 SYS    Systems                        34 MDF    MVSCOMP Defaults
  5 APG    ApplicationGroups              35 SDF    System Defaults
  6 APL    Applications                   36 ADF    Application Defaults
  7 EVT    Events                         37 AOP    Automation Operators
  8 SVP    Service Periods                38 NFY    Notify Operators
  9 TRG    Triggers                       39 NTW    Networks
 10 PRO    Processors                     40 XDF    Sysplex Defaults
 11 MTR    Monitor Resources              41 RES    Resident CLISTs
                                          42 SCR    Status Display
 13 PAC    Pacing Gates                   50 DMN    Remote Domains
                                          51 REF    Resource References

 20 PRD    Product Automation             99 UET    User E-T Pairs
 21 MSG    Messages
```

Adapting the System Name

You now have a basic PDB that is built from the sample add-on policy that is provided by the product.

About this task

In this policy, the default systems that are being automated are called SYS1, SYS2, and SYS3. These names have to be changed to match the names of your systems.

Procedure

1. Select 4 on the Option line and you see the systems that are listed as shown here:

```
------------------------------------------------------------------------------
                            Entry Name Selection                Row 1 from 3
Command ===>   ____________________________________________________  SCROLL===> CSR

Entry Type : System               PolicyDB Name   : TEST_PDB
                                  Enterprise Name : TEST_SYSTEMS

Action      Entry Name            Short Description
_________   SYS1                  System 1 of the SA Sample Sysplex
_________   SYS2                  System 2 of the SA Sample Sysplex
_________   SYS3                  System 3 of the SA Sample Sysplex
```

2. To rename the policy entry name of the system SYS1, enter r and press Enter. In the pop-up panel that is displayed next, enter the name of your system and press Enter again.

   The entry name is renamed, but one more renaming action is necessary.
3. Enter SI and press Enter.

   This action leads you to the **System Information** policy. Here again, you have to change the field **Image/System name** to match the name of your system. Before the change, the panel might look like as follows:

```
--------------------------------------------------------------------------------
                              System Information
Command ===>   ____________________________________________________________________

Entry Type : System                PolicyDB Name   : TEST_PDB
Entry Name : SYS1                  Enterprise Name : TEST_SYSTEMS

Operating system       : MVS
Image/System name. . . . SYS1___

The following specifications are for MVS systems only:
Primary JES. . . . . . . JES2____  Primary JES2/JES3 subsystem name
System monitor time. . . 00:59     Time between monitor cycles (hh:mm or NONE)
Gateway monitor time . . 00:15     Time between monitor cycles (hh:mm or NONE)
Automation table(s). . . INGMSG01_____________________________________________
```

4. Rename SYS1 here to your system name and press PF3 to leave the dialog box.

   You now see a group of messages that flow through the panel that shows the resources that are defined for your system. The message flow reflects the contents of the basic policy.
5. As a starting point, it is sufficient to automate just a single system. So you may leave SYS2 and SYS3 untouched and add further systems later on after the first system can be automated.
6. Press PF3 twice to return to the **Entry Type Selection** panel.

Adapting Application Job Names

Use the notes that you took during the planning exercise to change the default job names (where necessary) to the real job names.

Then, either delete or unlink those applications that are not used on the target system.

Select 6 on the Option line and press Enter. The **Entry Name Selection** panel for entry type Application is displayed.

```
                            Entry Name Selection              Row 1 from 31
Command ===>  _____________________________________________________ SCROLL===> CSR

Entry Type : Application           PolicyDB Name   : TEST_PDB
                                   Enterprise Name : TEST_ENTERPRISE

Action      Entry Name             C  Short Description
__________  AM                        Automation Manager
__________  AM2                       Spare Automation Manager
__________  APPC                      Advanced Peer-to-Peer Communication
__________  ASCH                      APPC Scheduler
__________  BLSJPRMI                  Build SNAP Tables for IPCS
__________  C_AM                   *  Class for Automation Manager Definitions
__________  C_APPL                 *  Class for general APL definitions
__________  C_JES2                 *  Class for Job Entry Subsystem 2
__________  DLF                       Data Lookaside Facility
__________  DSIRQJOB                 NetView JES-JobID-Requestor
__________  FFST                      First Failure Support Technology
__________  HSM                       Hierarchical Storage Manager
__________  IRRDPTAB                  RACF dynamic parse table loader
__________  JES2                      Job Entry Subsystem 2
__________  LLA                       Library Lookaside
__________  OAM                       Object Access Method
__________  OMPROUTE                  Open MVS MultiProtocol Routing Daemon
__________  OMVS                      Unix System Services subsystem
__________  RACF                      Resource Access Control Facility
__________  RESOLVER                  TCP/IP Name Resolver
__________  RMF                       Resource Measurement Facility
__________  RMFGAT                    RMF Monitor III Data Gatherer
__________  RRS                       Resource Recovery Services
__________  SYSVAPPL                  Automation Application
__________  SYSVIPLC                  IPL Data Gatherer
__________  SYSVSSI                   Automation Subsystem Interface
__________  TCPIP                     TCP/IP
__________  TSO                       Time Sharing Option
__________  VLF                       Virtual Lookaside Facility
AI________  VTAM                      Virtual Telecommunication Access Method
__________  ZFS                       z/OS File System
```

To change a job name for an application, enter AI next to that application and press Enter. A panel is shown as follows:

```
                        Application Information              Line 00000001
Command ===>  _____________________________________________________Scroll ===> PAGE

Entry Type : Application           PolicyDB Name   : TEST_PDB
Entry Name : VTAM                  Enterprise Name : TEST_ENTERPRISE

APL Type   : INSTANCE
Category . . . . . . . . . __________   (IBM-defined, user-defined or blank,
                                        see help)
Subcategory  . . . . . . . __________   (IBM-defined, user-defined or blank,
                                        see help)
Subsystem Name . . . . . . VTAM_______
Job Type . . . . . . . . . _________    (MVS NONMVS TRANSIENT)
Job Name . . . . . . . . . VTAM____
Transient Rerun  . . . . . ___          (YES NO)
Scheduling Subsystem . . . ________     (MSTR, JES Subsystem)
JCL Procedure Name . . . . __________________________
```

For example, if the VTAM job name is NET on the target system, change the value of the **Job Name** field in the panel appropriately. If you press PF3 twice, you return to the **Entry Name Selection** panel.

Follow the same steps if you want to enter the JCL Procedure Name.

To delete an application you do not need, enter D next to it and press Enter. You see a confirmation panel and press Enter again. However, if you want to use this application in the future, unlink it. The definitions are kept in the policy but the Customization Dialog does not create a resource for the application. You can link such an application any time later again.

To unlink an application you do not need, enter W next to it and press Enter. You notice that the application is linked to a group called BASE_SYS. Enter M (for reMove) next to it and press PF3.

Changing System Defaults

When you create the basic PDB the first time, you have no experience yet with customization and operations of the product. It is recommended to monitor what is going on to further familiarize yourself with the product, and then switch on automation.

About this task

The approach protects you from stumbling into pitfalls where unintended automation might happen by accident.

To do so, you can switch automation globally off by setting the **Automation** flag to LOG in the System Defaults (SDF). No automation takes place but the commands that the automation would run are shown in the netlog.

Procedure

1. Select 35 on the Option line and press Enter.

   You see a single system **SYSTEM_DEFAULTS** policy, similar to what is shown here:

```
------------------------------------------------------------------------------
                           Entry Name Selection               Row 1 from 1
Command ===> _____________________________________________SCROLL===> CSR

Entry Type : System Defaults       PolicyDB Name   : TEST_PDB
                                   Enterprise Name : TEST_ENTERPRISE

Action      Entry Name             Short Description
__________  SYSTEM_DEFAULTS        System Defaults
```

2. Under **Action**, specify AF and press Enter.

   You enter the **Automation Flag Processing** dialog that is shown here:

```
------------------------------------------------------------------------------
                        Automation Flag Processing
Command ===> ____________________________________________________________

Entry Type : System Defaults       PolicyDB Name   : TEST_PDB
Entry Name : SYSTEM_DEFAULTS       Enterprise Name : TEST_ENTERPRISE

Resource   : System Defaults

Line Commands: Exi (Exits), Dis (Disable Times)
Automation Level: YES, NO, LOG, EXITS

Cmd  Flag            Auto    Exits  DisableTimes

___  Automation (A)  LOG_
___  Initstart  (I)  ____
___  Start      (S)  ____
___  Recovery   (R)  ____
___  Terminate  (T)  ____
___  Restart   (RS)  ____
```

3. Change the value of Automation from **YES** to **LOG** and press PF3.

   No automation can happen accidentally. But do not forget to turn the flag back to YES after you are familiar with the product.
4. Press PF3 again until you are back on the initial panel, the primary panel, of the Customization Dialog.

Building the Configuration Files

You completed the steps to create a basic automation policy. You now create the configuration files (SOCNTL).

Procedure

1. Enter option 2 from the **IBM Z System Automation 4.4 Customization Dialog** to start the Build dialog.

```
                            Configuration Build
Option ===> ____________________________________________________________________

  1 Build a complete enterprise
  2 Build sysplex group or stand alone system
      Sysplex / System name. .  _____________________ (*, ?, or name)
  3 Build entry type or entry name
      Entry Type. . . . . . . . SDF_                  (*, ?, or type)
      Entry Name. . . . . . . . SYSTEM_DEFAULTS_____ (*, ?, or name)
  4 View build report

Build options:
  Output Data Set . . . . _____________________________________________________
  Mode. . . . . . . . . . ONLINE      (ONLINE BATCH)
  Type. . . . . . . . . . MODIFIED    (MODIFIED ALL)
  Configuration . . . . . NORMAL__    (NORMAL ALTERNATE TERTIARY)

Job statement information:  (used for BATCH build)
//AOFBUILD JOB ______________________________________________________________
//*___________________________________________________________________________
//*___________________________________________________________________________
```

2. The Configuration Assistant already created a SOCNTL file for you. So, in the **Output Data Set** field, enter the value that you specified for `sa_automation_policy` in the `Configuration Options` file and append `'.SOCNTL'`, surrounded by single quotation marks. For example: `'`*`USER`*`.POLICY.NAME.PDB.SOCNTL'`.
3. Change Type from MODIFIED to ALL.
4. Select Option **1 Build a complete enterprise** and press Enter.

   Messages are displayed and after a time, the build process completes successfully.

Results

You created an SOCNTL file from your basic policy that can be loaded on the target system. For the remaining steps, you need a console to enter system commands on the target system.

Starting the Automation Manager

The Automation Manager is started with a standard MVS Start command:.

Procedure

Issue: `S INGEAMSA,JOBNAME=AM,TYPE=COLD,SUB=MSTR`

Note: If you specified a different JCL procedure name (`sa_am_start_proc`) or job name (`sa_am_start_job.1`) for the Automation Manager, then use the values specified in the `INGDOPT Configuration Options` file.

Results

The Automation Manager initializes and issues the following message when the initialization is complete:

```
HSAM1308I SA z/OS PRIMARY AUTOMATION MANAGER INITIALIZATION COMPLETE,TYPE=COLD
```

If this message is not displayed, see which of these actions can help you:

- Find the messages that are displayed on the MVS console to identify the cause.
- Verify that you have the proper authority.
- Be sure that you performed correctly all steps of the configuration that are described above in this chapter.

Starting the Subsystem Interface Task

The Subsystem Interface Task is started with a standard MVS Start command:

Procedure

Issue: `S CNMSJ010,JOBNAME=SYSVSSI,SUB=MSTR`

Note: If you specified a different JCL procedure name (`sa_nvssi_start_proc`) or job name (`sa_nvssi_start_job`) for the subsystem interface task, then use the values as specified in the `INGDOPT Configuration Options` file.

Results

After the task is initialized, the following message appears:

```
CNM541I NetView subsystem SYSV is fully functional
```

If this message is not displayed, see which of these actions can help you:

- Find the messages that are displayed on the MVS console to identify the cause.
- Verify that you have the proper authority.
- Be sure that you performed correctly all steps of the configuration that are described above in this chapter.

Starting the Automation Agent

The Automation Agent is started with a standard MVS Start command.

Procedure

Issue: `S INGENVSA,JOBNAME=SYSVAPPL,SUB=MSTR`

Note: If you specified a different JCL procedure name or job name for the Automation Agent, then use the values (`sa_saagent_start_proc` or `sa_saagent_start_job`, respectively) as found in the `INGDOPT Configuration Options` file.

Results

After the Automation Agent is initialized up to the point where logging on is possible, it responds with the following message:

```
*002 DSI802A ING01 REPLY WITH VALID NCCF SYSTEM OPERATOR COMMAND
```

If this message is not displayed, see which of these actions can help you:

- Find the messages that are displayed on the MVS console to identify the cause.
- Verify that you have the proper authority.
- Be sure that you performed correctly all steps of the configuration that are described above in this chapter.

After this message is displayed, you are able to log on to the NetView 3270 console.

The Automation Manager instructs the Automation Agent to load the SOCNTL data set. When done, another message is displayed:

```
 HSAM1330I LOAD_ACF REQUEST COMPLETED SUCCESSFULLY ON SYS1.
AOF767I AUTOMATION OPTIONS: 729
 . STOP      - CANCEL AUTOMATION
 . PAUSE     - SUSPEND AUTOMATION
 . NOSTART   - DO NOT AUTOMATE SUBSYSTEM STARTUP
 . RUNMODE=x - SET RUNMODE (CURRENT *ALL)
 . ENTER     - CONTINUE
*003 AOF603D ENTER AUTOMATION OPTIONS OR 'R' (RE-DISPLAY) - DOMAIN ING01
```

What to do next

Press Enter to close this message.

Verification

When the Automation Manager and the Automation Agent are both started successfully, log on to the NetView console.

Procedure

1. To log on, enter `LOGON APPLID` (*domain*).

   For `domain`, use the value that you specified for `net_netview_domain_id` in the `INGDOPT Configuration Options` file. A panel is shown as follows:

```
   NN    NN                     VV         VV
   NNN   NN   EEEEEE  TTTTTTTT  VV       VV    II   EEEEEE  WW         WW  TM
   NNNN  NN   EE         TT      VV     VV     II   EE      WW    W    WW
   NN NN NN   EEEE       TT       VV   VV      II   EEEE     WW  WWW  WW
   NN  NNNN   EE         TT        VV VV       II   EE        WWWW WWWW
   NN   NNN   EEEEEE     TT         VVV        II   EEEEEE     WW   WW
   NN    NN                        V

   5697-NV6 © Copyright IBM Corp.      1986, 2014 - All Rights Reserved
  U.S. Government users restricted rights - Use, duplication, or disclosure
       restricted by GSA ADP schedule contract with IBM corporation.
          Licensed materials - Property of IBM Corporation
   Domain = ING01                    NV62 SA42 NM

          OPERATOR ID ==>            or LOGOFF
             PASSWORD ==>
              PROFILE ==>            Profile name, blank=default
            HARDCOPY LOG ==>         device name, or NO, default=NO
  RUN INITIAL COMMAND ==>            YES or NO, default=YES
     Takeover session ==>            YES, NO, or FORCE, default=NO



          Enter logon information or PF3/PF15 to logoff
```

2. For **OPERATOR ID**, specify 0PER1. For the **PASSWORD**, specify 0PER1.

   The entries are default credentials that are set up for you to get into SA z/OS initially.

   Note: Secure the environment as soon as possible following the guidelines in Chapter 11, “Security and Authorization,” on page 151.
3. After you log on, press Enter, when you see message: =X= *** DSI662I SCREEN HELD.
4. Enter INGAMS on the command line for the operational command INGAMS.

   A panel as follows is then displayed:

```
INGKYAM0                  SA z/OS  - Command Dialogs       Line  1    of 2
 Domain ID   = IPUFL     ---------- INGAMS   ----------    Date = 03/20/22
 Operator ID = JMH             Sysplex = SYS1PLEX          Time = 12:34:25

 Cmd:  A Manage       B Show Details  C Refresh Configuration  D Diagnostic

 CMD System   Member    Role  Status     Sysplex  XCF-Group Release Comm E2E
 --- -------- --------- ----- ---------- -------- --------- ------- ---- ---
     SYS1     SYS1      AGENT READY      SYS1PLEX INGXSG    V4R3M0  XCF
     SYS1     SYS1$$$$1 PAM   READY      SYS1PLEX INGXSG    V4R3M0  XCF
```

   The statuses of the Primary Automation Manager (PAM) and of the Automation Agent are READY.

Chapter 10. Traditional SA z/OS Configuration

This information describes the tasks required to configure SA z/OS components on the SA z/OS host systems. Included is information on configuring SA z/OS on both focal point and target systems.

The target system configuration does not require some of the steps used for the focal point configuration. Any configuration step that does not apply to the target systems is indicated. Many of the configuration steps have corresponding planning activities and explanations in the introductory planning sections. Chapter 12, “Configuring SA z/OS Workstation Components,” on page 185 describes installation on workstations.

In the information, 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 document, 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 LPALSTxx” on page 74 and “Step 4E: Update LNKLSTxx” on page 74.

Overview of Configuration Tasks

The major tasks required for configuring SA z/OS on a focal point are listed in Table 9 on page 67.

Table 9. Configuration Tasks for SA z/OS Host Systems. √=Required, *=Optional

Task	SysOps	ProcOps
Step 1: SMP/E Installation. Refer to Chapter 8, “SMP/E Installation,” on page 47	√	√
“Step 2: Allocate System-Unique Data Sets” on page 69	√	√
“Step 3: Allocate Data Sets for the ISPF Dialog” on page 72	√	√
“Step 4: Configure SYS1.PARMLIB Members” on page 73	√	√
“Step 5: Configure SYS1.PROCLIB Members” on page 76	√	√
“Step 6: Configure NetView” on page 78	√	√
“Step 7: Preparing the Hardware” on page 84	√	√
“Step 8: Preparing the VM PSM” on page 92		*
“Step 9: Configure the Automation Manager” on page 95	√	

Table 9. Configuration Tasks for SA z/OS Host Systems. ✓=Required, *=Optional *(continued)*		
Task	**SysOps**	**ProcOps**
“Step 10: Configure the Component Trace” on page 97	✓	
“Step 11: Configure the System Logger” on page 97	*	
“Step 12: Configure ISPF Dialog Panels” on page 98	✓	✓
“Step 13: Verify the Number of available REXX Environments” on page 103	✓	✓
“Step 14: Configure Function Packages for TSO” on page 103	*	
“Step 15: Configure Alert Notification for SA z/OS” on page 104	*	
“Step 16: Compile SA z/OS REXX Procedures” on page 108	*	*
“Step 17: Defining Automation Policy” on page 108	✓	✓
“Step 18: Define Host-to-Host Communications” on page 109	✓	✓
“Step 19: Enabling SA z/OS to Restart Automatic Restart Manager Enabled Subsystems” on page 110	✓	
“Step 20: Define Security” on page 111	✓	✓
“Step 21: Configure the Status Display Facility (SDF)” on page 111	*	*
“Step 23: Check for Required IPL” on page 117	✓	✓
“Step 22: Configure System Automation Info Broker” on page 112	*	
“Step 24: Automate System Operations Startup” on page 118	✓	✓
“Step 25: Verify Automatic System Operations Startup” on page 119	*	
“Step 26: Configure USS Automation” on page 120	*	
“Step 27: Configure and Run the System Automation Data Store” on page 120	*	
“Step 28: Configure Db2 as an alternative database of dynamic resources” on page 123	*	
“Step 29: Configure and Run the System Automation Operations REST Server” on page 125	*	
“Step 30: Configure the Policy Services Provider” on page 135	*	
“Step 31: Enable the End-to-End Automation and Connect an SAplex to IBM Z Automation Web Console” on page 140	*	
“Step 32: Copy and Update Sample Exits” on page 140	*	*
“Step 33: Install Relational Data Services (RDS)” on page 140	*	
“Step 34: Install CICS Automation in CICS” on page 141	*	
“Step 35: Install IMS Automation in IMS” on page 143	*	
“Step 36: Install ZWS Automation in ZWS” on page 144	*	
“Step 37: Configuring GDPS” on page 146	*	
“Step 38: Installing Tivoli Enterprise Portal Support” on page 148	*	

Step 2: Allocate System-Unique Data Sets

SysOps	ProcOps
√	√

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
- 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
√	

The data sets in Table 10 on page 69 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 NetView in “Step 5: Configure SYS1.PROCLIB Members” on page 76.

Table 10. 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	Partitioned	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	Library	DSILIST
Contains the members to be used when testing the automation table.	INGALLC0	Partitioned	DSIASRC
Stores the output report produced from running tests of the automation table.	INGALLC0	Partitioned	DSIARPT
Contains VTAM source definitions for the sample network.	INGALLC0	Partitioned	DSIVTAM

Table 10. Data Sets for Each Individual Automation Agent (continued)

Purpose	Sample job to allocate the data set	Organization	DD name in the NetView startup procedure
NetView log data sets	INGALLC0	VSAM	DSILOGP, DSILOGS
NetView trace data set	INGALLC0	VSAM	DSITRCP, DSITRCS
DVIPA Workload Statistics	INGALLC0	Sequential	CNMDVIPP, CNMDVIPS
NetView save/restore data set	INGALLC0	VSAM	DSISVRT

Step 2B: Data Sets for Automation Agents

SysOps	ProcOps
✓	

The data sets in Table 11 on page 70 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 NetView in “Step 5: Configure SYS1.PROCLIB Members” on page 76.

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
Automation status file	INGALLC2	VSAM	AOFSTAT
Dump file for diagnostic information	INGALLC2	Sequential	INGDUMP

The data set in Table 12 on page 70 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 NetView in “Step 5: Configure SYS1.PROCLIB Members” on page 76.

Table 12. 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 2C: Data Sets for Automation Managers (Primary Automation Manager and Backups)

SysOps	ProcOps
✓	

The data sets in Table 13 on page 71 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 5: Configure SYS1.PROCLIB Members” on page 76.

Each subplex requires one separate set of the following:

- The schedule override file
- The configuration information data set
- The automation manager takeover file

Table 13. 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 14 on page 71 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 5: Configure SYS1.PROCLIB Members” on page 76.

Table 14. 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 15 on page 72 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.

These files also need to be referred to in the automation manager startup procedure in “Step 5: Configure SYS1.PROCLIB Members” on page 76.

Table 15. 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 2D: SA z/OS Password Store Data Set

SysOps	ProcOps
√	√

The data set in Table 16 on page 72 is required once per SAplex and can be shared between automation agents. It needs to be referred to in the startup procedure for each automation agent's NetView in “Step 5: Configure SYS1.PROCLIB Members” on page 76.

Table 16. Shared Data Set for Each SAplex			
Purpose	**Sample job to allocate the data set**	**Organization**	**DD name in the NetView startup procedure**
Password data set for INGPW	INGALLC4	VSAM	INGPSWD

Step 3: Allocate Data Sets for the ISPF Dialog

About this task

SysOps	ProcOps
√	√

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

For system operations and processor operations, these data sets include:

- The ISPF table library data set that contains the values you enter in the customization dialog
- The SA z/OS configuration file: this is the output data set for the customization dialog when building the SA z/OS configuration.

Data Set Name	Purpose
ING.CUSTOM.AOFTABL	ISPF table output library for the customization dialog
ING.CUSTOM.SOCNTL	SA z/OS configuration files

Note:

- Make a note of these data set names. They are used in “Step 12: Configure ISPF Dialog Panels” on page 98. If you rename the data sets, you need to adapt the corresponding names in that step.
- As the ISPF dialog makes use of the ISPF service QUERYENQ, the ISPF SHOW_ENQ_DISPLAY option must NOT be set to NO in the ISPF Configuration Table. For more Information, refer to the *ISPF Planning and Customization* manual.

Step 4: Configure SYS1.PARMLIB Members

SysOps	ProcOps
√	√

The *xx* suffix on each SYS1.PARMLIB data set member can be any two characters chosen to match your IEASYS naming scheme. 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 about how to achieve this.

Step 4A: Update IEAAPF*xx*

About this task

SysOps	ProcOps
√	√

Define authorized libraries to the authorized program facility (APF) in an IEAAPFxx member.

Edit the IEAAPF*xx* member to add the following to the APF:

- ING.SINGLOAD, ING.SINGLINK, ING.SINGLPA

Step 4B: Update SCHED*xx*

About this task

SysOps	ProcOps
√	

Sample: INGESCH

If you run z/OS 2.1 or higher then you can skip this step since all automation-related components are already part of the z/OS-delivered Program Property Table.

Otherwise consult the chapter "SCHEDxx", sub-chapter "Program Property Table" in *z/OS MVS Initialization and Tuning Reference* to find out, which of the entries listed in sample member INGESCH are already part of the z/OS-provided PPT. Edit the SCHEDxx member to ensure that it includes all the missing statements for INGESCH.

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), and the automation manager to run without being swapped out of memory.

Step 4C: Update MPFLST*xx*

About this task

SysOps	ProcOps
√	√

Sample: INGEMPF

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 108). Using the customization dialog you can obtain a list of the messages that are 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.

In addition SA z/OS provides a sample member called INGEMPF in the SINGSAMP sample library. This contains the IDs of all of the messages that occur in the INGMSGSA NetView automation table that is delivered with SA z/OS. Thus if you concatenate both the INGEMPF member and the dynamically-created MPFLSTSA member, you obtain a list of all of the messages that are used in the INGMSGSA and INGMSG01 automation tables.

Alternatively, update the content of your MPFLST*xx* member based on INGEMPF and INGMSGSA, and make sure that all of the messages that are listed there are forwarded to automation.

Note: GDPS clients should also review appropriate GDPS documentation for MPFLSTxx recommendations.

Step 4D: Update LPALST*xx*

About this task

SysOps	ProcOps
√	√

Edit the LPALST*xx* member to add ING.SINGLPA 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.SINGLPA 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. For a complete description of dynamic LPA and PROGxx, see *z/OS MVS Initialization and Tuning Reference*.

Notes:

1. Make sure that the SA z/OS load library is cataloged in the master catalog, or copy the members in ING.SINGLPA to a data set that is in the master catalog.
2. Be sure you do not have any data sets containing load modules with prefixes of AOF, ISQ, ING, or HSA in these members.
3. If ING.SINGLPA is to be placed in SYS1.PARMLIB member LPALST*xx*, ensure the data set organization is of type PDS.

Step 4E: Update LNKLST*xx*

About this task

SysOps	ProcOps
√	√

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

Add SINGLOAD (recommended) and SINGLINK (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.

Edit the LNKLSTxx member to add the following to the LNKLST concatenation: ING.SINGLOAD, ING.SINGLINK.

You can avoid an IPL: You can also code a PROGxx member to add libraries to the LNKLST concatenation. If you do this, no IPL is required. For a complete description of dynamic LSTLNK and PROGxx, see *z/OS MVS Initialization and Tuning Reference*.

Step 4F: Update BPXPRM*xx*

The zFS dataset SINGZFS contains the USS related parts of SA z/OS. It must be mounted to enable UNIX automation through SA z/OS.

Add the content of INGEBPX member, which resides in the SA z/OS sample library SINGSAMP, to your BPXPRM*xx* concatenated member.

You can dynamically mount the SINGZFS dataset without an IPL: Issue the MVS command `D OMVS` to get the current BPXPRM*xx* member concatenation. Use `T OMVS` command to activate the changed definitions of the updated BPXPRM*xx* members.

Step 4G: Update IEFSSN*xx*

SysOps	ProcOps
√	√

Sample: INGESSN

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 started task names. The four-character prefix that you specify must match the four-character prefix of the NetView started task names. For example, if you specify SYSV, the names of the NetView job name must be SYSV*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. Please adapt the content of your IEFSSNxx member accordingly. If you run NetView 5.x then define:

```
SUBSYS SUBNAME(SYSV)        /* NETVIEW-SA SUBSYSTEM NAME             */
```

  If you run NetView 6.x, then define:

```
SUBSYS SUBNAME(SYSV) /* NETVIEW-SA SUBSYSTEM NAME             */
  INITRTN(DSI4LSIT)
```

- To prevent JESx from starting before SA z/OS during the IPL process, indicate that in your IEFSSNxx member accordingly.

```
SUBSYS SUBNAME(JES2)        /* JES2 IS THE PRIMARY SUBSYSTEM  NAME    */
   PRIMARY(YES) START(NO)
```

  However if you plan to start JESx before NetView, remove the START(NO) option from your definitions in the IEFSSNxx member. For the correct syntax of your environment check the *z/OS MVS Initialization and Tuning Reference*.

Step 4H: Update JES3IN*xx*

About this task

SysOps	ProcOps
√	

Sample: INGEJES3

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 4I: Update SMFPRM*xx*

About this task

SysOps	ProcOps
*	

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)
```

For the correct syntax of your environment check the *z/OS MVS Initialization and Tuning Reference*.

Step 5: Configure SYS1.PROCLIB Members

SysOps	ProcOps
√	√

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

Step 5A: NetView Startup Procedures

About this task

SysOps	ProcOps
✓	✓

- **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:

  – Ensure that the PPIOPT parameter is set to PPI. Several SA z/OS functions use PPI communication as a base, for example, USS automation.

- **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).

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

  – 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 69.

  Note: Adaptation of the JCL procedure names to meet the four-character prefix defined in the IEFSSnxx member will be done in "Step 24: Automate System Operations Startup" on page 118 when defining the jobnames for Automation NetView.

  If you do not make ING01 your domain name, make a note of what your NetView domain name is. This information is needed for system operations. See also *IBM Z System Automation Defining Automation Policy* for more information on enterprise definitions.

  See *IBM Z NetView Installation: Configuring Additional Components* for further details about how to modify the NetView startup procedure.

Step 5B: Startup Procedures Required for System Operations Only

About this task

SysOps	ProcOps
✓	

- **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 all systems where System Automation will be installed and run.

  Configure that copy to your needs. In particular, make sure that the DD concatenations mentioned in "Step 2: Allocate System-Unique Data Sets" on page 69 include the data sets that you allocated there. In addition, consider configuring the following point:

  – 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 9: Configure the Automation Manager" on page 95):

```
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 69.

- **Other System Operations Startup Procedures**

  Copy the following members from the SINGSAMP data set to members of the SYS1.PROCLIB of all systems where System Automation will be installed and run:

  HSAPIPLC
  This procedure gathers IPL statistics and stores the information in the IPLDATA file. Once set up, you can view sysplex-wide IPL data with the command `INGPLEX IPL`.

  You can give the procedure any name.

  It is recommended that you define this procedure in your automation policy as an application with the option 'START ON IPL ONLY'.

  Alternatively, you can start this procedure during every IPL. This can be accomplished by adding `COM='S HSAPIPLC,SUB=MSTR'` to a COMMAND*xx* parmlib member that is shared by all systems in the sysplex.

  INGPHOM
  This procedure is used internally by SA z/OS to process sysplex data for CF paths.

  The procedure name must *not* be changed.

  INGPIPLC
  This procedure is used internally by SA z/OS to compare IPL data.

  The procedure name must *not* be changed.

  INGPIXCU
  The procedure is used internally by SA z/OS to process sysplex data for Sysplex utilities (for example, Couple Data Set management, Coupling Facility management, and so on.). Once set up, you can view and manage related Sysplex CDS and CF data with the commands `INGPLEX CDS` and `INGPLEX CF`.

  The procedure name must *not* be changed.

  Follow the configuration 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 169.

- *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 6: Configure NetView

SysOps	ProcOps
√	√

This section discusses how to configure several aspects of NetView:

Step 6A: Configure NetView DSIPARM Data Set

SysOps	ProcOps
√	√

Sample: INGSTGEN

A sample is provided for this step in the INGSTGEN member of the SINGSAMP library. Copy the contents of INGSTGEN to your C*xx*STGEN or C*xx*STUSR and configure it to match your installation. See the INGSTGEN sample for further details.

Copy any DSIPARM and SINGPARM member that you need to configure 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.

GDPS®
This enables GDPS to run under SA z/OS. Use this definition regardless of the specific GDPS product that is running (GDPS Metro, GDPS 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 Metro (formerly named GDPS/PPRC)

HM
For GDPS HM (formerly named GDPS/PPRC HM)

XRC
For GDPS XRC (formerly named GDPS/XRC)

GM
For GDPS GM (formerly named 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.

GDPSSAT
This statement enables the GDPS Satellite support required in a GDPS Continuous Availability (GDPS AA) environment.

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 *IBM Z NetView Installation: Configuring Additional Components*.

Timer Catchup Processing: SA z/OS requires `init.TIMER=NO` for its timer catchup processing. If you do not have any timers defined in the SA z/OS policy or none of the defined timers has the `CATCHUP=YES` option, you can code `init.TIMER=YES` to cause your saved timers to be restored at NetView startup time.

Refer to the NetView documentation for details about configuring 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.

If you want to define your own synonyms, you may use INGSYNU member which is automatically included from AOFMSGSY. By using this member, you can avoid changing the product supplied AOFMSGSY member.

Copy and edit the AOFMSGSY member that resides in ING.SINGPARM 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 Z System Automation 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 INGSYNU
     │
     ├── %INCLUDE INGMSGU1
     │
     ├── %INCLUDE INGMSG02
     │
     └── %INCLUDE INGMSGU2
```

Message Revision Table

During the build of the automation control file, a NetView Revision Table is being built by the customization dialog. For more information about activating the built Message Revision Table (MRT), see the topic "AT/MRT/MPF Build and Activate" in *IBM Z System Automation User's Guide*.

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.SINGPARM. Uncomment the following parameters and specify your values:

FUNCTIONLEVEL
System Automation user function level. You can specify a value up to 5-digit number in the range 0 - 65535. The default value is 0.

This parameter is used to control what capabilities the agent is allowed to perform when operating in a System Automation sysplex environment with systems at different service levels.

Refer to Appendix G, “Function levels,” on page 225 for a detailed description and when to use function levels.

GRPID
2-byte XCF group ID. Default is blank. The value must be the same as specified for GRPID in the corresponding member HSAPRMxx.

PLEXID
2-byte suffix used to build the extended XCF communication group with the name INGPX$xx. For suffix xx you can specify all alphanumeric characters except the suffix $$ (#@ are acceptable, for instance). If PLEXID is not specified, the automation agent is not a member of the extended XCF communication group.

If you are running more than one NetView on your system, they cannot be in the same extended XCF communication group. Either leave the PLEXID commented out or make sure that the PLEXIDs used for the NetViews on the same system are unique.

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 parameter defines if the NetView agent should establish a connection to the system logger at initialization time. If NO is specified the agent does not establish a connection. The default is YES which causes the agent to connect to the following log streams:

- HSA.WORKITEM.HISTORY
- HSA.MESSAGE.LOG

You may also specify the value GRPID. This allows you to have separate log streams per subplex. If GRPID is specified and the value of the GRPID keyword is not blank, the agent connects to the following log streams:

- HSA.GRP*xx*.WORKITEM.HISTORY
- HSA.GRP*xx*.MESSAGE.LOG

If the value of the GRPID keyword is blank the LOGSTREAM keyword defaults to YES.

Note: Both values, LOGSTREAM and GRPID, must be the same as in the PARMLIB member HSAPRM*xx* that is used to start the related automation manager.

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 Z System Automation End-to-End Automation*.

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,LIFECYCLE=500,LOGSTREAM=YES
```

An example of an invalid syntax is:

```
GRPID = 34 , LIFECYCLE = 500
```

Step 6B: Modifying NetView DSIPARM Definitions for an Automation Network

SysOps	ProcOps
√	√

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.

In the AOFOPFGW member for each system, define the operator IDs used for both outbound and inbound gateway autotasks.

For example, in the example inbound and outbound gateway autotasks, the gateway autotask definitions in AOFOPFGW on domain CHI01 are as follows:

```
GATCHI01 OPERATOR
PROFILEN AOFPRFAO
GATATl01 OPERATOR
PROFILEN AOFPRFAO
GATATl02 OPERATOR
PROFILEN AOFPRFAO
```

Step 6C: Configure NetView for Processor Operations

About this task

SysOps	ProcOps
	√

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 180.

Processor operations uses automation table entries for its operation. The required AT entries are shipped as part of the SysOps automation table INGMSGSA and are activated if the ProcOps TOWER statement is specified.

ISQMSGU1

This empty member is supplied by processor operations and is included in the SysOps automation table INGMSG01. 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.

ProcOps user AT ISQMSGU1 is included in INGMSG01, so it runs in parallel to the ProcOps AT entries which are shipped by System Automation.

Step 6D: Configure the NetView Message Translation Table

About this task

SysOps	ProcOps
*	

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 configured. (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 6E: Add the REXX Function Packages to DSIRXPRM

About this task

SysOps	ProcOps
√	√

All NetView REXX functions of SA z/OS are packaged in module INGRXFPG. This package will be automatically loaded to the function package table in NetView at initialization time.

If you plan to use CICSplex System Manager REXX API, then specify the following line in your NetView stylesheet:

```
REXX.FUNCPKGLIST.SYS.EYU9AR00=EYU9AR00
```

Step 7: Preparing the Hardware

SysOps	ProcOps
√	√

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 22.

In addition, refer to the publications *Hardware Management Console Guide* and *Support Element Operations Guide* for details about your HMC and SE.

Note that the following hardware preparation steps are based on SE or HMC user interface (UI) style 'Tree Style'. It is recommended to change to 'Tree Style' using the console's User Settings task in case the 'Classic Style' is also supported by your console.

Step 7A: Preparing the HMC

Enable the HMC API, Set the SNMP Community Names and the SNMPv3 Information

About this task

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:

Procedure

1. For this task, you need to log on in *Access Administrator* mode on your HMC.
2. Select **HMC Management** task overview.
3. From the **Configuration** task sub-list, select **Customize API Settings**. Make sure the **Enable SNMP APIs** check box is set in the Customize API Settings window.

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.

4. 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.

   A CPC is controlled over the SNMP interface when it is configured for connection protocol SNMP, using the Processor (CPC) entry in the SA z/OS Customization Dialog. See “Step 12: Configure ISPF Dialog Panels” on page 98 and “Step 17: Defining Automation Policy” on page 108 for further details on maintaining theSA z/OS Policy Database.

Note: The Customize API Settings window must be open.

5. 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:

Parameter	Description
Name	Specify the name in uppercase with a maximum length of 8 characters. . Record this name and use it when you go to define the processor entry for the CPC in your SA z/OS policy database 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 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. In order to support SNMPv3 protocol, providing more security and traffic encryption, the *SNMPv3 User Information* must be defined. Then, you can use SA z/OS ProcOps to manage CPCs defined on the HMC and to execute CPC HW commands over SA z/OS ProcOps SNMP interface.

A CPC is controlled over the SNMP interface using SNMPv3 protocol when it is configured for connection protocol SNMP and SNMPv3 is enabled at the Processor (CPC) entry in the SA z/OS Customization Dialog. See "Step 12: Configure ISPF Dialog Panels" on page 98 and "Step 17: Defining Automation Policy" on page 108 for further details on maintaining the SA z/OS Policy Database.

Note: The Customize API Settings window must be open.

8. For a new ProcOps SNMPv3 User, select the **SNMPv3 Users** table **Add** push button. In the **SNMPv3 User Information** data entry window, enter the following information:

Parameter	Description
User Name	Specify an SNMPv3 user name. The user name must be at least 8 characters in length and cannot exceed 31. **Note:** HMC allows specification of user names of length 32, but the SA is limited to a support maximum of 31 characters for SNMPv3 Users.
Password	Specify a password for the SNMPv3 user. The password must be at least 8 characters in length and cannot exceed 31. **Note:** HMC allows specification of passwords length 32, but the SA is limited to support maximum of 31 characters for the SNMPv3 passwords
Access Type	Select the **Read/Write** radio button.

9. Select the **OK** push button to save the changed settings and close the data entry window.
10. If you have finished the SNMP API settings, select the **Apply** push button of the Customize API Settings window to save the changes.
11. 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 and ProcOps SNMP ISQET32 Redirection

Procedure

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 **HMC Service Management** task overview.
3. Select **Console Internal Code**.
4. Uncheck the **Block Automatic Licensed Internal Code Change Installation** check box.
5. Press **Save** to make the change permanent.

Results

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

About this task

Depending on the processor hardware interfaces, the CPCs that are to be managed must be defined to the HMC. For the SA z/OS BCP internal interface, the master HMC, which must have the 'Licensed Internal Code Change Installation' enabled, is used as a router between the CPC where SA z/OS is running, and other targeted CPCs.

For the System Automation ProcOps TCP/IP SNMP connection, the HMC serves as a single point of control. Alternatively, System Automation ProcOps SNMP can be configured to communicate directly with a CPC over TCP/IP or the BCP Internal Interface, by addressing its Support Element.

For detailed information about 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 7B: Preparing the SE

Enable the SE API, Set the Community Name and the SNMP Information

About this task

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:

Procedure

1. For this task, you need to log on in *Access Administrator* mode on your Support Element.

   Note: You can log on to the SE from an HMC where your CPC is defined by using the Systems Management task, then select your CPC and choose the Recovery task to perform the Single Object Operations action.
2. Select **SE Management** task overview.
3. From the **Configuration** task sub-list, select **Customize API Settings**. Make sure the **Enable SNMP APIs** and the **Allow Capacity Change API Request** check boxes are both set in the Customize API Settings window.

   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.
4. **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 over TCP/IP or BCP Internal Interface redirection. A CPC is controlled over the SNMP interface when it is configured with connection protocol SNMP in the Processor (CPC) entry of the SA z/OS Customization Dialog. See “Step 12: Configure ISPF Dialog Panels” on page 98 and “Step 17: Defining Automation Policy” on page 108 for further details on maintaining the SA z/OS Policy Database.
5. a) **Note:** 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:

Parameter	Description
Name	Specify the name in uppercase with a maximum length of 8 characters. Record this name and use it when you are going to specify the processor entry for the CPC in your SA z/OS policy database with connection type SNMP. **Note:** For a SNMP over BCP Internal Interface connection, this name requires 127.0.0.1 to be defined in the Address field. In case you want to switch between SNMP over TCP/IP and SNMP over BCP Internal Interface, two separate entries with this community name are required. The TCP/IP SNMP entry should provide the IP address of the ProcOps focal point system. The BCP Internal Interface entry must provide the Support Element loopback address 127.0.0.1.
Address	Use the IP address of your SA z/OS ProcOps focal point system. **Note:** For a SNMP over BCP Internal Interface connection, this address must be 127.0.0.1, the loopback address of the Support Element.
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. 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. **Note:** For a SNMP over BCP Internal Interface connection, the mask must be 255.255.255.255.
Access Type	Select the **Read/Write** radio button.

b) **Note:** 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:

Parameter	Description
Name	Specify the name in uppercase with maximum length of 8 characters. Record this name and use it when you go to define the processor entry for the CPC in your SA z/OS policy database with connection type INTERNAL.
Address	The required address is 127.0.0.1
Network Mask	The required address 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. In order to support SNMPv3 protocol, providing more security and traffic encryption, the *SNMPv3 User Information* must be defined. Then, you can use SA z/OS ProcOps to manage CPCs defined on the HMC and to execute CPC HW commands over SA z/OS ProcOps SNMP interface.

   A CPC is controlled over the SNMP interface using SNMPv3 protocol when it is configured for connection protocol SNMP and SNMPv3 is enabled at the Processor (CPC) entry in the SA z/OS Customization Dialog. See "Step 12: Configure ISPF Dialog Panels" on page 98 and "Step 17: Defining Automation Policy" on page 108 for further details on maintaining the SA z/OS Policy Database.

   Note: The Customize API Settings window must be open.
8. For a new ProcOps SNMPv3 User, select the **SNMPv3 Users** table **Add** push button. In the **SNMPv3 User Information** data entry window, enter the following information:

Parameter	Description
User Name	Specify an SNMPv3 user name. The user name must be at least 8 characters in length and cannot exceed 31. **Note:** SE allows specification of user names of length 32, but the SA is limited to a support maximum of 31 characters for SNMPv3 Users.
Password	Specify a password for the SNMPv3 user. The password must be at least 8 characters in length and cannot exceed 31. **Note:** SE allows specification of passwords length 32, but the SA is limited to support maximum of 31 characters for the SNMPv3 passwords.
Access Type	Select the **Read/Write** radio button.

9. Select the **OK** push button to save the changed settings and close the data entry window.
10. If you have finished the API settings, select the **Apply** push button of the Customize API Settings window to save the changes.
11. 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

About this task

This task is only required if you use the BCP internal interface protocol to monitor and control the CPC processor and its partitions. For this task, you need to log on in *System Programmer mode* on your CPC's Support Element. To complete this task:

Procedure

1. Select the **Systems Management** task set.
2. Select the **CPC Operational Customization** sub-task list.
3. Choose the **Change LPAR Security** selection. The Change Logical Partition Security window is displayed showing the security settings from the active IOCDS for the logical partitions defined on this CPC.
4. For each listed logical partition that should use the BCP internal interface to control other partitions on this CPC or other CPCs in the same processor LAN, check the **Cross Partition Authority** check box and save the settings.

Enabling Capacity Change API Requests

About this task

To be able to perform capacity changes (for example, CBU) using the SA z/OS hardware interfaces BCPii or SNMP ProcOps, the 'Allow Capacity Change API requests' flag must be set:

Procedure

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 **Allow Capacity Change API Requests** check box is set in the Customize API Settings window.

Step 7C: Setting IBM Z BCPii Permissions (IBM z14 or later)

There are two levels of BCPii permissions settings, system (processor, CPC) and partition (LPAR) settings. One end of the BCPii communication is the System Automation instance, running in a partition (z/OS LPAR) of a system, and the other end can be any system or partition that is specified as target for the BCPii commands. The Support Elements on both ends must be attached to an IBM processor network, which allows to establish a session between them. The BCPii permission settings on both ends must enable this communication.

System BCPii Permissions

Use the Hardware Management Console (HMC) task **System Details** of the system and select the tab **Security** to validate or change the **System BCPii Permissions** setting. This setting enables the system to accept BCPii query and action commands from all or selected partitions only. This setting affects the targeted system only, not its partitions.

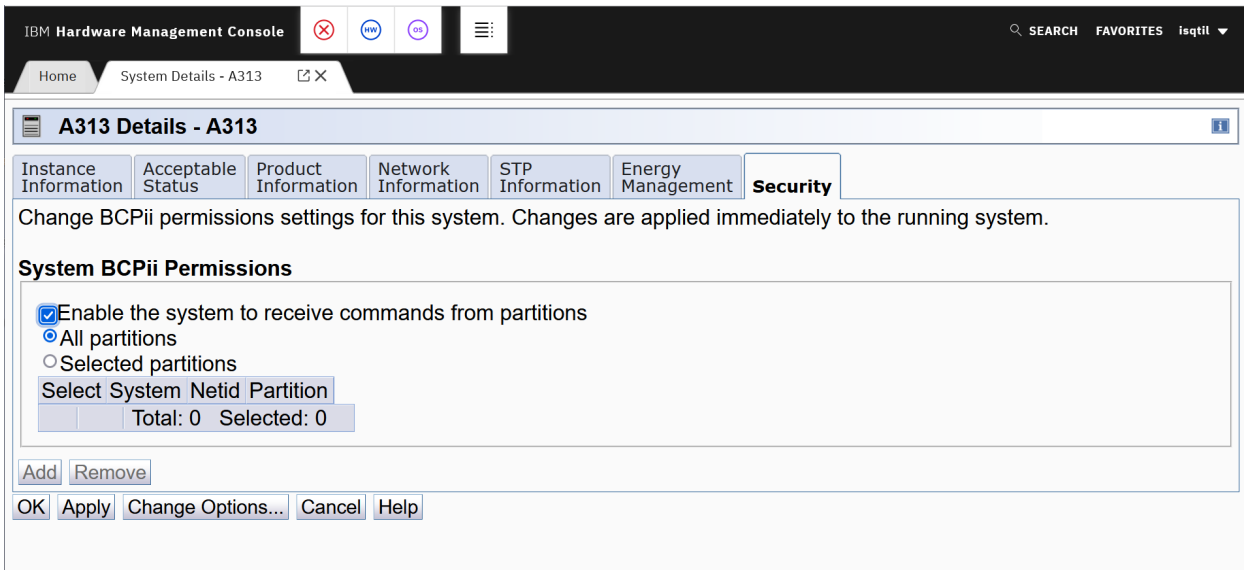


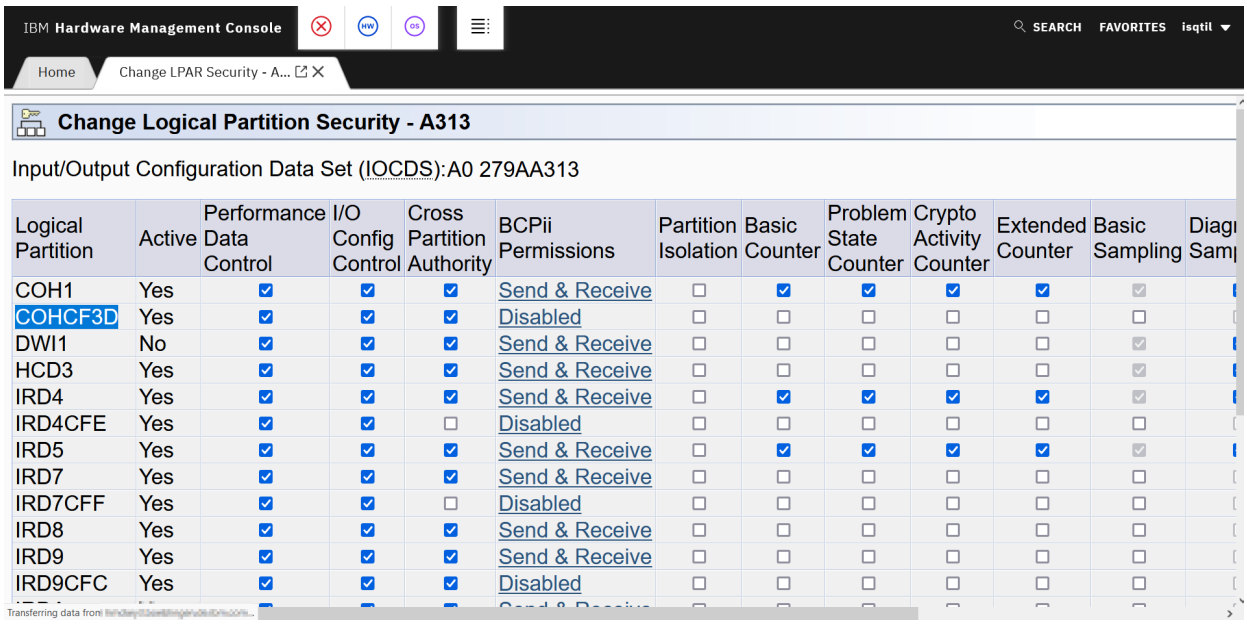
Figure 7. System BCPii Permissions settings

The IBM supplied default is to allow the CPC to receive commands from all partitions of any connected system in your IBM processor network. Note that the SA-BCPii function does not require to change this default!

If you need to change the IBM **System BCPii Permissions** default, make sure that defined processors in your policy database are enabled to receive BCPii commands from all partitions defined in your policy database, which need to establish SA-BCPii connections to that processor. This is mandatory for all ProcOps ISQET32 and GDPS INTERNAL SA-BCPii connections.

LPAR BCPii Permissions

Use the HMC task **Change LPAR Security** in section **Operational Customization** of the system to validate or change the BCPii permission settings on LPAR level to enable receiving incoming query and action commands and to enable sending BCPii commands from the selected partition to other processors and partitions. The IBM supplied default for each partition is to have BCPii send and receive disabled.



Logical Partition	Active	Performance Data Control	I/O Config Control	Cross Partition Authority	BCPii Permissions	Partition Isolation	Basic Counter	Problem State Counter	Crypto Activity Counter	Extended Counter	Basic Sampling	Diagı Samı
COH1	Yes	☑	☑	☑	Send & Receive	☐	☑	☑	☑	☑	☑	
COHCF3D	Yes	☑	☑	☑	Disabled	☐	☐	☐	☐	☐	☐	
DWI1	No	☑	☑	☑	Send & Receive	☐	☐	☐	☐	☐	☑	
HCD3	Yes	☑	☑	☑	Send & Receive	☐	☐	☐	☐	☐	☑	
IRD4	Yes	☑	☑	☑	Send & Receive	☐	☑	☑	☑	☑	☑	
IRD4CFE	Yes	☑	☑	☐	Disabled	☐	☐	☐	☐	☐	☐	
IRD5	Yes	☑	☑	☑	Send & Receive	☐	☑	☑	☑	☑	☑	
IRD7	Yes	☑	☑	☑	Send & Receive	☐	☐	☐	☐	☐	☐	
IRD7CFF	Yes	☑	☑	☐	Disabled	☐	☐	☐	☐	☐	☐	
IRD8	Yes	☑	☑	☑	Send & Receive	☐	☐	☐	☐	☐	☐	
IRD9	Yes	☑	☑	☑	Send & Receive	☐	☐	☐	☐	☐	☐	
IRD9CFC	Yes	☑	☑	☑	Disabled	☐	☐	☐	☐	☐	☐	

Figure 8. LPAR BCPii Permissions settings

If you want to use the SA-BCPii protocol for hardware automation, you must change the IBM default LPAR BCPii permission setting from disabled to either send, receive, or both, depending on the role of the partition in your System Automation policy database or the partition type. For example, CF or SSC partitions cannot issue BCPii commands and do not require BCPii send enabled.

If you have enabled the LPAR to receive BCPii commands, you must also specify whether you allow to receive commands from all partitions in your processor network, or give permission only to selected processors and partitions.

If the BCPii permissions setting includes send, you must check the **Cross Partition Authority** checkbox, which is displayed next to the **BCPii Permissions** setting for the selected partition.

It is your responsibility to apply BCPii permissions that allow local and cross CPC BCPii communication that match with your System Automation policy definitions. If you have processors and LPARs defined in your policy database, System Automation cannot determine the current BCPii permission settings, before it tries to access the partition.

Step 7D: 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.
- In addition, the well-known port for SNMP, port 161, must be configured to allow TCP packet traffic between ProcOps and the SE/HMC.

Step 8: Preparing the VM PSM

SysOps	ProcOps
	*

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

About this task

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*.SINGMENV — 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 93.

If these steps are processed successfully then the PSM can be started.

Configuration

Procedure

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 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 for IPv4 connections, or in the preferred conventional form for IPv6 connections. If a node name is specified and that node name has several addresses, all addresses that are returned are used.

Note node names cannot be used to validate IPv6 connections and are ignored if the PSM is running an IPv6 environment.

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
  * Normal focal point Netview IPv6
FD00:9:152:40:840:FFFF:80:253
  /* the backup IPv6
  FD00:9:152:40:840:FFFF:80:254
```

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, PSMIPV4, and CLEANUP 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.

PSMIPV4
You should set this keyword to Y to indicate that PSM should enforce IPv4 sockets in an IPv6-enabled environment. Supported values are Y or N. If PSMIPV4 is not specified, default value N is used by the PSM and IPv6 will be preferred.

CLEANUP
The number of days to retain logger files. The default is 0, which means that old ISQLOG files will not be removed. The valid range is 0 to 365 days. If the value is specified, all logger files older than CLEANUP days will be automatically removed from A-disk.

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 9: Configure the Automation Manager

SysOps	ProcOps
✓	

Step 9A: XCF Characteristics

SA z/OS uses XCF characteristics with any communication method. Ensure that transport classes for CLASSLEN(956) and CLASSLEN(4028) are defined. An XCF group name should not be assigned to the transport classes.

When setting up the sysplex you need to be aware that SA z/OS has a maximum XCF message length of 3500 bytes. You can either use an existing transport class with the appropriate class length, or define a new transport class.

Step 9B: Configuring 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 specify the HSAPLIB DD statement in the automation manager startup procedure member.

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 69. Refer to Appendix C, “Syntax for HSAPRM00,” on page 211 for the contents of this sample and the description of the parameters.

Step 9C: ARM Instrumentation of the Automation Manager

The automation manager can be enabled for Automatic Restart Manager (ARM). However, this is optional and not recommended if you use the *BASE best practice policy.

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 9D: Security Considerations

The started task that invokes the automation manager (see INGEAMSA in the sample library) must have the following access rights:

1. If the automation manager is to be started with option BLOCKOMVS=YES the started task must be defined by RACF as a superuser for UNIX System Services. For more information about BLOCKOMVS refer to Appendix C, “Syntax for HSAPRM00,” on page 211.
2. If you are not a superuser, you must have access to the OMVS segment.
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
- The schedule override file
- The configuration information file (DDname HSACFGIN)
- The takeover file

Step 10: Configure the Component Trace

About this task

SysOps	ProcOps
√	

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 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 11: Configure the System Logger

SysOps	ProcOps
*	

Although 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.

Notes:

1. The LOGSTREAM parameter in the HSAPRMxx parmlib member is set to YES by default. The automation manager connects to the logger address space at initialization.
2. If you set the LOGSTREAM parameter to NO, no access is established to the system logger. “Step 11: Configure the System Logger” on page 97 is then unnecessary.
3. If you set the LOGSTREAM parameter to GRPID, the automation manager connects to the logger address space at initialization time. However, the log streams to which the automation manager connects depend on the value of the GRPID parameter. For more information, see Appendix C, “Syntax for HSAPRM00,” on page 211.

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 HSAJD*xxx* members as input and make the changes accordingly.

Note: Do not change the provided MAXBUFSIZE values in the HSAJDxxx job. The provided values match the size of the expected data.

For a sysplex environment, you must additionally add the log structures to the CFRM policy:

```
STRUCTURE   NAME(HSA_LOG)
            SIZE(17408)
            FULLTHRESHOLD(0)
            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 17408 K (17M). If the size specified was not large enough, you may see message IXL015I STRUCTURE ALLOCATION INFORMATION. Since System Logger manages the space of the structure, there is no need for additional monitoring. The parameter FULLTHRESHOLD(0) disables XES monitoring and potential IXC585E messages.

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 about 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 12: Configure ISPF Dialog Panels

SysOps	ProcOps
√	√

SA z/OS ships the following type of ISPF dialogs:

- For defining automation policy: The customization dialog is used to create system operations and processor operations configuration and automation definitions.

These ISPF dialogs are 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 D, “INGDLG Command,” on page 217 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 Z System Automation User's Guide*.

Step 12A: Allocate Libraries for the Dialogs

SysOps	ProcOps
✓	✓

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

The recommended way to start the customization dialog is Alternative 1. SA z/OS provides a sample INGEDLG in the SINGSAMP library for this.

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 *ISPF User's Guide*.

Remember: Throughout this step use the names of the data sets that you created in “Step 3: Allocate Data Sets for the ISPF Dialog” on page 72.

Alternative 1: Dynamic Allocation using INGDLG

About this task

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 Appendix D, “INGDLG Command,” on page 217 for the use of INGDLG to allocate libraries.

Alternative 2: Allocation of the libraries as part of the TSO Logon Procedure

About this task

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.SINGPENU,DISP=SHR
             DD …

//ISPMLIB    DD …
             DD DSN=ING.SINGMENU,DISP=SHR
             DD …
```

```
//ISPSLIB    DD …
             DD DSN=ING.SINGSENU,DISP=SHR
             DD …

//ISPTLIB    DD …
             DD DSN=ING.CUSTOM.AOFTABL,DISP=SHR  1
             DD DSN=ING.SINGTENU,DISP=SHR
             DD …

//ISPLLIB    DD …
             DD DSN=ING.SINGLOAD,DISP=SHR
             DD …

//SYSPROC    DD …
             DD DSN=ING.SINGTREX,DISP=SHR
             DD …

//AOFTABL    DD DSN=ING.CUSTOM.AOFTABL,DISP=SHR  1

//AOFPRINT   DD SYSOUT=…  2

//AOFIPDB    DD DSN=ING.SINGPDB,DISP=SHR  3
```

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 applications, 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 temporarily hold file tailoring output and to hold the JCL for batch jobs. See *z/OS ISPF Planning and Customizing* for more information on the ISPCTL1 data set.
2. The ellipses (...) in the DD statements indicate the presence of more information in the JCL: for example, other data sets in a concatenation.
3. 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.
4. The AOFTABL DD statement (1) is required to store ISPF tables created when you use the customization dialog. Such tables are used, for example, during pdb import or when the administrator modifies the SA z/OS policy definitions from the SA z/OS customization dialog. 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 ISPF Dialog” on page 72).
5. 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.

6. The AOFIPDB DD statement (3) points to the SA z/OS sample library.

   The AOFIPDB DD statement is required for using best practice policies and for building system operations configuration files.

7. 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.

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 12B: Logging Modifications to Data Set

About this task

During APAR apply, a log of the modifications is created and it is written to that data set. If the data set does not exist a dynamic allocation is attempted using a default name. If this name does not fit the installation's naming conventions, or a data set allocation is not allowed at all, this data set should be pre-allocated. Besides the APAR apply, this data set is needed by the report functions which are invoked by the "Report Selection Menu".

Hint: The Report Output Data Set is required for APAR apply. For more information about this data set, refer to "How to Apply Service Updates" in *IBM Z System Automation Defining Automation Policy*.

Step 12C: Invoking the ISPF Dialogs

SysOps	ProcOps
✓	✓

The ISPF dialogs are invoked with the INGDLG command. Parameters of this command determine which set of dialogs is invoked (that is, system operations or processor operations).

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 9 on page 101 is an example of what a modified panel might look like.

```
-----------ISPF APPLICATION SELECTION MENU--------------------------------
OPTION ===> ____________________________________________________

  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
  T  TUTORIAL   - Display information about ISPF/PDF
  X  EXIT       - Terminate ISPF using log and list defaults

Enter END command to terminate ISPF.
```

Figure 9. ISPF Application Selection Menu

The option for the customization dialog 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.

There are two alternatives to invoke the ISPF dialog:

- "Using INGDLG" on page 102. This is the recommended method.
- "Using TSO Logon or Your own Automation Procedure" on page 102.

Using INGDLG

About this task

If you let INGDLG, described in Appendix D, "INGDLG Command," on page 217, 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.SINGTREX(INGDLG)'' +
  ''HLQ(MYHLQ)                       +
    AOFTABL(ING.CUSTOM.AOFTABL)      +
    SELECT(ADMIN)'')'
```

Alternatively, you can invoke the dialogs using TSO REXX execs:

```
/* REXX ADMIN */
ADDRESS ISPEXEC "SELECT CMD(EXEC 'ING.SINGTREX(INGDLG)'" ,
"'HLQ(ING)                        " ,
/* HLQ is the hlq of the SMP/E output data sets */
" AOFTABL(ING.CUSTOM.AOFTABL)     " ,
" SELECT(ADMIN)                  )"
```

A sample member called INGEDLG is provided in SINGSAMP sample library for invocation of INGDLG with data set allocation done by INGDLG.

Using TSO Logon or Your own Automation Procedure

About this task

This is the example to be followed if you allocated the data sets using the TSO logon procedure or an automation procedure 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))'

T,'PGM(ISPTUTOR) PARM(ISR00000)'
:
X,'EXIT'
*,'?' )
&ZTRAIL = .TRAIL
)END
```

Step 12D: Verify the ISPF Dialog Installation

About this task

SysOps	ProcOps
√	√

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. On the Customization Dialog Primary Menu that will appear, verify the release in the panel header

Step 13: Verify the Number of available REXX Environments

About this task

SysOps	ProcOps
√	√

Change the value of the maximum number of available REXX environments to at least 2000. 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 2000. 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 24.

Step 14: Configure Function Packages for TSO

SysOps	ProcOps
*	

This step is only required when you intend to use one of the following features:

- the general purpose command receiver
- ZWS/OPC Command Receiver for I/F
- the syntax checking for automation table overrides
- command INGRCRDX

Step 14A: Installation of the TSO REXX Function Package INGTXFPG

About this task

Add INGTXFPG to the function package table in the appropriate TSO module below. TSO/E provides the following samples in SYS1.SAMPLIB that you can use to code your load modules:

Table 17. TSO Load Modules for INGTXFPG

Sample name	Load module name
IRXREXX1	(IRXPARMS for MVS)
IRXREXX2	(IRXTSPRM for TSO/E)
IRXREXX3	(IRXISPRM for ISPF)

There are various considerations for providing your own parameters modules. For further details, see the chapter "Function Package" of the *TSO REXX Reference*. The different considerations are based on whether you want to change a parameter value for an environment(s) initialized:

- for ISPF
- for both TSO/E and ISPF sessions
- in a non-TSO/E address space

Select the appropriate sample parameters modules, for example **IRXREXX2 for TSO/E and batch PGM=IKJEFT01** and make the highlighted and underlined changes similar to the example both:

```
PACKTB_SYSTEM_FIRST DC A(PACKTB_ENTRIES)        /* Address of the first*/
*                                               /* System Entry        */
```

```
PACKTB_SYSTEM_TOTAL DC F'3'                    /* Total number of     */
*                                              /* system entries      */
PACKTB_SYSTEM_USED DC F'3'                     /* Number of System    */
*                                              /* entries in use      */
PACKTB_LENGTH DC F'8'                    /* Length of each PACKTB entry */
PACKTB_FFFF DC X'FFFFFFFFFFFFFFFF'       /* Set the PACKTB end marker   */
PACKTB_ENTRIES EQU *                     /* System Package Table entries */
PACKTB_ENTRY_MVS EQU *                    /* The MVS-PACKTB             */
PACKTB_NAME_MVS DC CL8 'IRXEFMVS'        /* 1. Set function package name */
PACKTB_NAME_MVS DS 0C                    /* Point to the next entry     */
PACKTB_ENTRY_TSO EQU *                    /* The TSO PACKTB entry       */
PACKTB_NAME_TSO DC CL8 'IRXEFPCK'        /* 2. Set function package name */
PACKTB_NEXT_TSO DS 0C                    /* Point to the next entry     */
PACKTB_ENTRY_SAM EQU *                   /* The SAM PACKTB entry        */
PACKTB_NAME_SAM DC CL8 'INGTXFPG'        /* 3. Set SA function package  */
PACKTB_NEXT_SAM DS 0C                    /* Point to next entry         */
```

Procedure

1. Link-edit the REXX default parameters module with the corresponding names. For example, the load module for the sample IRXREXX2 must have the name IRXTSPRM.
2. Place the resultant REXX default parameter module in the LPALST.
3. Make sure that the function package INGTXFPG resides in the LinkList.

Step 14B: Install SA Provided Authorized TSO Command INGPAUTH

System Automation delivers the authorized TSO command INGPAUTH. The Relational Data Services require that the TSO command INGPAUTH must be defined as an authorized command in TSO. This can be achieved by adding the command name to the PARMLIB member IKJTSOxx in SYS1.PARMLIB under AUTHCMD.

Use the TSO/E command PARMLIB UPDATE(xx), or MVS command SET IKJTSO=xx, to activate the new settings. Be sure that INGPAUTH is concatenated in the LINKLIST.

Refer to “Accessing authorized TSO command INGPAUTH” on page 172 and complete further SAF relevant actions that secure the infrastructure appropriately.

Step 15: Configure Alert Notification for SA z/OS

SysOps	ProcOps
*	

This section describes the configuration steps that are required for alert notification by SA z/OS.

In order to use alert notification the following must apply to the affected resource in your automation policy:

1. The inform list of the resource must contain at least one of the following communication methods (it can also be defaulted or inherited):
   - IOM: via the IBM Tivoli System Automation for Integrated Operations Management (SA IOM) peer-to-peer protocol
   - EIF: via a Tivoli Event Integration Facility (EIF) event
   - TTT: via XML
   - USR: via a user-defined alert handler
2. Codes must be present on the reserved message ID, INGALERT, that are suitable for the chosen communication methods.

For full details about the installation of related workstation components, refer to Chapter 12, “Configuring SA z/OS Workstation Components,” on page 185. Additionally for further information, see *IBM Z System Automation Defining Automation Policy* and *IBM Z System Automation Customizing and Programming*.

Furthermore, for each system that is able to trigger an alert (that is, to issue an INGALERT command), the ALERTMODE parameter must be set to the chosen communication methods with the INGCNTL command, for example:

```
INGCNTL SET ALERTMODE='IOM EIF TTT USR'
```

You can also use the following command to set alerting for all available communication methods:

```
INGCNTL SET ALERTMODE=ON
```

The available communication methods are:

- IOM: via the SA IOM peer-to-peer protocol
- EIF: via EIF events
- TTT: via XML
- USR: via a user-defined alert handler

Depending on the chosen communication methods, additional customization is required. This is described in the following sections. Note that you can combine the INGCNTL calls shown in this section in one single invocation.

For more details about INGCNTL, see *IBM Z System Automation Programmer's Reference*.

Enabling Alert Notification via SA IOM Peer-To-Peer Protocol

About this task

On each system that can connect to an SA IOM server you must set the host name and port number with INGCNTL, for example:

```
INGCNTL SET ALERTHOST=IOMSRV1:1040
```

For more details about INGCNTL, see *IBM Z System Automation Programmer's Reference*.

Enabling Alert Notification via EIF Events

About this task

Alert notification uses the message adapter or the confirmed message adapter service of the event/automation service (E/AS) component of NetView to create EIF events and to integrate SA z/OS and products such as IBM Tivoli Netcool® OMNIbus (OMNIbus).

On each system that is able to send EIF events you must set the PPI receiver name of the E/AS with INGCNTL, for example:

```
INGCNTL SET EIFPPI=INGEVOMN
```

To enable the confirmed message adapter service for EIF events, you have to set the confirm parameter for each system with INGCNTL, too. For example:

```
INGCNTL SET CONFIRM=EIF
```

For more details about INGCNTL, see *IBM Z System Automation Programmer's Reference*. For more details about the differences between the two message adapter services, see *IBM Z NetView Customization Guide*.

Starting the Event/Automation Service

About this task

The E/AS and the steps to enable it are described in the chapter, "Setting Up UNIX System Services for the NetView Program" in *IBM Z NetView Installation: Configuring Additional Components*. The following section only provides additional information about how to enable the NetView message adapter service or the confirmed message adapter service of E/AS for alert notification.

The E/AS can be started either with a job from an MVS system console, or from a UNIX System Service command shell. In either case, startup parameters must be provided in the form of the following initialization files:

- Global initialization file (Default: IHSAINIT)
- Message adapter configuration file (Default: IHSAMCFG)
- Confirmed message adapter configuration file (Default: IHSANCFG)

The sample startup JCL IHSAEVNT for E/AS is located in NETVIEW.CNMSAMP. The initialization and configuration files are assumed to be located in a data set that is allocated to the DD name IHSSMP3. Perform the following updates to the sample to meet the requirements of your installation:

Procedure

1. If you do not use the default name IHSAINIT for the global initialization file, pass the name of your file via the parameter INITFILE.
2. Pass the name of your file via the appropriate parameter:
   - for the message adapter service configuration file, use the MSGCFG parameter if you do not use the default file name IHSAMCFG.
   - for the confirmed message adapter service configuration file, use the CMSGCFG parameter if you do not use the default file name IHSANCFG.
3. In the DD statement, specify the data set names of your installation.

Configuring the Global Initialization File

Procedure

1. Make sure that the NetView message adapter service or the confirmed message adapter service is also started when you start the E/AS. This is done by commenting out one of the following statements:
   - for the message adapter service:

     ```
     NOSTART TASK=MESSAGEA
     ```

   - or for the confirmed message adapter service:

     ```
     NOSTART TASK=MESSAGEC
     ```

   The other services are not needed by alert notification, so prevent them from starting.
2. Specify INGEVOMN, or any other name of the PPI receiver ID, in the following statement:

   ```
   PPI=INGEVOMN
   ```

   You can also pass the PPI receiver ID as a parameter when starting E/AS. Make sure that you define the same name you specified with INGCNTL.

Configuring the NetView Message Adapter Service

About this task

Configuration of the NetView message adapter service is done in the message adapter configuration file, as follows:

Procedure

1. Provide the IP address or host name and, optionally, the port address of the event receiver. This can be virtually any kind of server that can handle EIF events but SA z/OS supplies integration with IBM Tivoli Netcool OMNIbus (OMNIbus).
2. Specify the name of the NetView message adapter format file. The version of this file that is to be used by alert notification is delivered in ING.SINGSAMP(INGMFMTO). If this is in its own data set (RECFM VB LRECL 516), copy it to a data set that is concatenated to IHSSMP3.

Configuring the NetView Confirmed Message Adapter Service

About this task

Configuration of the NetView confirmed message adapter service is done in the confirmed message adapter configuration file, as follows:

Procedure

1. Provide the IP address or host name and, optionally, the port address of the event receiver. This can be virtually any kind of server that can handle EIF events and sends a reply by receiving the event. SA z/OS supplies integration with IBM Tivoli Netcool/OMNIbus (OMNIbus).
2. Specify the name of the NetView confirmed message adapter format file. The version of this file that is to be used by alert notification is delivered in INGSINGSAMP(INGMFMTO). If this is in its own data set, copy it to a data set (RECFM VB LRECL 516) that is concatenated to IHSSMP3.

Enabling Alert Notification via XML

About this task

Alert notification can help with creating trouble tickets automatically. Thus SA z/OS collects details about the failed resource and stores it in a details data set. It also creates XML data with overview information.

You must use the INGCNTL command to set the host name and port number to send the XML data to on each system that is able to create trouble ticket information, for example:

```
INGCNTL SET TTTHOST=TDISRV1:8000
```

You must also specify allocation data for the details data set, for example:

```
INGCNTL SET TTTDATA='ING.TTT.DATA 1 1'
```

For more details, see INGCNTL in *IBM Z System Automation Programmer's Reference*.

Enabling Alert Notification via User-Defined Alert Handler

About this task

SA z/OS allows you handle an alert in any other way that you choose.

On each system that is able to run a user-defined alert handler you must specify the command to be executed with INGCNTL, for example:

```
INGCNTL SET USRHANDLER=MYHANDLER
```

For more details about INGCNTL, see *IBM Z System Automation Programmer's Reference*.

For details about the parameters that are passed and the return codes, see the sample handler delivered in ING.SINGSAMP(AOFEXALT).

Step 16: Compile SA z/OS REXX Procedures

SysOps	ProcOps
*	*

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.SINGREXX 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.SINGREXX members. Modify the data set names and jobcard in INGEREXR as necessary and submit the job. The ING.SINGREXX.CREXX library can be modelled on ING.SINGREXX, and ING.SINGREXX.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.

Notes:

1. A compiler return code of 4 can be expected and is acceptable.
2. SA z/OS has *not* been tested to run with the REXX Alternate Library. Officially, this is not a supported environment.
3. The NOTESTHALT compiler option should not be used when compiling System Automation REXX.

Step 17: Defining Automation Policy

About this task

SysOps	ProcOps
√	√

Before you can start using automation, you need to define your automation policy using the customization dialog.

If you start from scratch:

Procedure

1. Use the IBM best practice policies that are delivered with SA z/OS, *BASE and any others as required, and create your new policy database. Read the information in the section "Creating a New Policy Database" in *IBM Z System Automation Defining Automation Policy*.

2. Next adjust and extend your automation policy. Start by working with the following policy objects:
   - Applications
   - Application groups
   - Monitor Resources
   - Processors
   - Systems
   - A group for each sysplex

Results

You can find detailed information about how to perform these steps in *IBM Z System Automation 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

About this task

IBM recommends that you use the SA z/OS best practice sample policies to define your SA z/OS components. When you have defined the policies for the SA z/OS components, use the BUILD command to create the configuration files. The BUILD command is available from various panels of the customization dialog. For more information about how to perform this step, refer to *IBM Z System Automation Defining Automation Policy*. You can use the sample job INGEBBLD in the SINGSAMP sample library to create the configuration files in batch.

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 configuration files manually. A manually edited automation control file may damage your automation.

Step 17B: Distribute System Operations Configuration Files

About this task

You need to make the configuration 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 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
√	√

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: Configure VTAM Connectivity

About this task

SysOps	ProcOps
√	

The configuration of VTAM bases on Mode table and Major Nodes.

Consult the description of member INGEMTAB that resides in the SINGSAMP sample library. INGEMTAB generates appropriate VTAM mode tables for the NetView application. Incorporate the resulting mode tables into your SYS1.VTAMLIB concatenation of your active VTAM startup procedure.

SA z/OS provides a sample major node member INGENET that resides in the SINGSAMP sample library. Adapt and rename INGENET according to your needs and incorporate it into your SYS1.VTAMLST concatenation of your active VTAM startup procedure.

Notes:

1. 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. It is expected that OMEGAMON is installed and has been configured for VTAM. See *IBM Z NetView Installation: Configuring Additional Components* and *z/OS Communications Server: SNA Network Implementation Guide* for more details.
2. The NetView primary program operator interface task (PPT) is defined as AUTH=(NVSPACE,SPO). This causes unsolicited VTAM messages to be broadcast on the SSI and thus available to NetView. If however you have another NetView, defined as a primary program operator application program (PPO), it receives unsolicited first and messages do not reach the NetView that is defined as a secondary program operator application program (SPO). See *IBM Z NetView Installation: Getting Started* for information on PPO and SPO definitions.

Step 19: Enabling SA z/OS to Restart Automatic Restart Manager Enabled Subsystems

About this task

SysOps	ProcOps
√	

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.SINGLINK 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 Z System Automation Defining Automation Policy*.

Step 20: Define Security

SysOps	ProcOps
√	√

Perform this step 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.

Refer to the Chapter 11, “Security and Authorization,” on page 151 and complete all SAF relevant actions that secure the infrastructure appropriately.

For a basic security setup consult the following subchapters:

- “Authorization of the Started Procedures” on page 152
- “Roles” on page 154
- “Operators” on page 155
- “Commands” on page 156
- “Use of Commands Cross System” on page 157
- “Use of Commands from TSO or Batch” on page 163
- “Resources” on page 165

See also the following sections in Chapter 11, “Security and Authorization,” on page 151 for other security options:

- “Other Security Options” on page 168
- “Securing Focal Point Systems and Target Systems” on page 169
- “Granting NetView and the STC-User Access to Data Sets” on page 169
- “Restricting Access to INGPLEX and INGCF Functions” on page 173
- “Restricting Access to Joblog Monitoring Task INGTJLM” on page 174
- “Security considerations to control Db2 subsystems” on page 174
- “Security for IBM Tivoli Monitoring Products” on page 175 (OMEGAMON)
- “Controlling Access to the Processor Hardware Functions” on page 180
- “Establishing Authorization with Network Security Program” on page 183

Note: To plan your RMTCMD-based INGSEND security, see the discussion of RMTCMD security features in the NetView documentation.

Step 21: Configure the Status Display Facility (SDF)

SysOps	ProcOps
*	*

If you decide to use SDF as the SA z/OS fullscreen operator interface for monitoring automated resource statuses at the NetView 3270 console, configuring SDF involves defining the following:

- SDF initialization parameters. These are defined in the AOFINIT member of a NetView DSIPARM data set.
- Copy and configure member INGPTOP in the ING.SINGPARM library concatenate it in the DSIPARM data set before the SA z/OS libraries. Configure it the system and sysplex names.

- Define and configure the following variables in the NetView style sheet depending on your environment. The sample below shows the definitions for an SDF focal point:

```
COMMON.AOF_AAO_SDFROOT.0 = 3
COMMON.AOF_AAO_SDFROOT.1 = &SYSNAME
COMMON.AOF_AAO_SDFROOT.2 = SYS1 SYS2
COMMON.AOF_AAO_SDFROOT.3 = SYSA SYSB SYSC
```

  To share the definitions across all systems, the recommendation is as follows:

  – Stem variable *COMMON.AOF_AAO_SDFROOT.0* defines how many consecutive variables are incorporated. Set the variable to 1 for each system that is NOT the SDF focal point system. And set the variable to the index of the last compound variable that defines the root names being monitored by the SDF focal point system.
  – Stem variable *COMMON.AOF_AAO_SDFROOT.1* should always specify the system symbol that resolves to the current system.
  – Stem variables *COMMON.AOF_AAO_SDFROOT.n* with N > 1 specify all root names being monitored by the SDF focal point system. Each variable can specify one or more root names. You may group the root names like in the sample above by the sysplex membership or by other criteria. Duplicate root names are ignored.

  You may use other panel/tree members than the default members AOFPNLS and AOFTREE for some root names like:

```
SYS1 SYS2/MYPNLS
SYSA&SLASH./MYTREE SYSB/MYPNLS2/MYTREE SYSC
```

  For SYSA, the panel member defaults to AOFPNLS. For SYS2, the tree member defaults to AOFTREE. For SYS1 and SYSC, both default members are being used.

  Notes:

  1. Both default members must not contain any variable that is subject to replication (See "AOFTREE" and "Status Component Panel Definition" in *IBM Z System Automation Programmer's Reference*).
  2. NetView interprets two consecutive slashes as the beginning of a line comment. For this reason the sample above uses the symbol for the slash character followed by the slash character itself.

- Ensure that the inform list in the customization dialog contains SDF for the resources that you want to monitor (consider using, for example, system and sysplex defaults).
- Color and priority assignments for resource status types. These have default values that are set up by SA z/OS (see *IBM Z System Automation 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 System Information Panel of the customization dialog. If you do not specify a new root name, it defaults to the value specified for SYSNAME.

See "Customizing the Status Display Facility" in *IBM Z System Automation Customizing and Programming* for detailed information about customizing SDF.

Step 22: Configure System Automation Info Broker

SysOps	ProcOps
*	

System Automation Info Broker is an optional component of Z System Automation. You can use this component to forward SDF messages and user-defined messages from NetView environment into a Kafka® topic that's created on the Kafka Server before. Any Kafka consumer, for example, Automation Dashboards for Z Automation Web Console, can consume the forwarded messages.

Prerequisites

Ensure that the following prerequisites are met before you can configure and run the System Automation Info Broker:

- Java Runtime Environment (JRE) 17 or higher installed on z/OS.

  Starting from APAR OA65704, Java 17 or higher is required. If you have previously used the Info Broker with an earlier Java version, follow the migration steps in .
- z/OS UNIX System Services (USS) available with mounted USS file system for use by the System Automation data store.
- Apache Kafka has been installed, configured, and is running.

  Note: Apache Kafka is not part of Z System Automation. It must be downloaded and installed before using the System Automation Info Broker. For more information, see: https://kafka.apache.org/

  .

Steps

- "Step 22A: Set up the USS Directory Structure" on page 113
- "Step 22B: Define a Data Services Task (DST)" on page 113
- "Step 22C: Create a New Initialization Member in DSIPARM" on page 114
- "Step 22D: Set up the JCL Procedure and the Environment File" on page 115
- "Step 22E: Customize Properties of the System Automation Info Broker" on page 116
- "Step 22F: Start and Stop the System Automation Info Broker" on page 116

Step 22A: Set up the USS Directory Structure

After SMP/E installation of System Automation, there is a USS directory structure for System Automation Info Broker.

- /usr/lpp/ing/infobroker/bin

  This is the read-only directory that contains scripts.
- /usr/lpp/ing/infobroker/lib

  This is the read-only directory that contains the executables.
- /usr/lpp/ing/infobroker/config

  This is the read-only directory that contains the configuration samples.

Procedure

Create your own working directory to your needs. Assume that you have created the following custom directories.

- `/etc/ing/infobroker/config` - Configuration files are placed here.
- `/var/ing/infobroker/data` - This directory is used to store temporary data that should not be lost during IPL.

Next step

"Step 22B: Define a Data Services Task (DST)" on page 113

Step 22B: Define a Data Services Task (DST)

The Data Services Task (DST) that is defined for the System Automation Info Broker receives and forwards messages from the NetView environment to the System Automation Info Broker that runs in USS. These

following TASK statements in the NetView style sheet member cause a new Data Services Task (DST) to be started every time NetView is started. The DST itself starts the "System Automation Info Broker Hub" process, which is running as Java process in USS, when needed.

Procedure

Add the following four TASK.* statements into your NetView style sheet member, for example, *USER_ID*.USER.DSIPARM(CNMSTUSR). You can also find these four statements as example in the AOFSTYLE member, which can be browsed by **BR AOFSTYLE**. You can also copy these four lines in the section "Info Broker Hub" and paste to your NetView style sheet member.

```
TASK.INGTKDST.MOD=DSIZDST
TASK.INGTKDST.MEM=INGKINIT
TASK.INGTKDST.PRI=4
TASK.INGTKDST.INIT=Y
```

Note: Do not pre-concatenate an adapted version of the AOFSTYLE member.

Next step

“Step 22C: Create a New Initialization Member in DSIPARM” on page 114

Step 22C: Create a New Initialization Member in DSIPARM

An initialization member is required by the Data Services Task (DST) to connect to the System Automation Info Broker and for the Broker to know on which Kafka topic the messages are sent. A sample of the initialization member is provided in the INGKINIT member of the SINGPARM library.

Procedure

Copy the sample member INGKINIT to your environment DSIPARM PDS and adapt the following parameters to your needs. .

Parameter	Description	Default
DEBUG	Use DEBUG=ON to generate debug messages.	OFF
PIPENAME	Use PIPENAME to specify the pipe's full path.	`/tmp/inginfobroker.pipe`
CCSID	Use CCSID to specify the encoding of the data on the host. It's used to convert to UTF-8.	CCSID=1047
PROCNAME	Use PROCNAME to specify the procedure in PROCLIB used to start the Kafka Producer address space.	INGEKPRC
HEARTBEAT	Use HEARTBEAT to specify the heart beat interval in seconds.	60
TOPICSDF	Use TOPICSDF to specify the topic name to be used for SDF events.	IBM-SYSAUTO-SDF

Next step

“Step 22D: Set up the JCL Procedure and the Environment File” on page 115

Step 22D: Set up the JCL Procedure and the Environment File

The sample procedure ING.SINGSAMP(INGEKPRC) starts the Java application of System Automation Info Broker. The //STDENV statement points to an environment file that resides in your custom directory. This environment file is needed to set up the USS environment for the System Automation Info Broker Publisher - the Java process that is started by the NetView DST and forwards the messages to your Kafka server.

Procedure

1. Copy the sample procedure ING.SINGSAMP(INGEKPRC) to your PROCLIB and change it to the defaults used in your environment. In case you rename this example procedure to your needs, make sure to change the **PROCNAME** attribute in the DSIPARM initialization parameter (see “Step 22C: Create a New Initialization Member in DSIPARM” on page 114).
2. Modify the **CONFIG** parameter in the sample procedure to the custom directory of your choice.
3. Copy the sample environment file from the directory `/usr/lpp/ing/infobroker/config/ing.infobroker.environment`to your custom directory, for example, `/etc/ing/infobroker/config/ing.infobroker.environment`
4. Customize the variables in the environment file as instructed by the descriptions within that sample file.

System Symbol Support for System Automation Info Broker (OA62518)

This feature resolves system symbols in the configuration variables at startup. This enables the user to have a generic configuration file that acts as a "single source of truth" and can be shared across systems and sysplexes. The specific configuration for each system can then be controlled via system symbols.

By default, this feature is disabled. To enable this feature, navigate to the file '`ing.infobroker.environment`' and set the following property to true.

```
# Enable system symbol support [true/false].
ENABLE_SYSTEM_SYMBOL_SUPPORT=true
```

After the feature is enabled, you can use system symbols in '`ing.infobroker.environment`', '`ing.infobroker.properties`', and '`ing.infobroker.security.properties`' files.

Example 1, ing.infobroker.environment:

```
IBRKR_CONFIG_HOME=/etc/&SYSNAME./ing/infobroker
```

On SYS1, this setting resolves to *IBRKR_CONFIG_HOME=/etc/SYS1/ing/infobroker*.

On SYS2, this setting resolves to *IBRKR_CONFIG_HOME=/etc/SYS2/ing/infobroker*.

Example 2, ing.infobroker.properties:

```
ing.infobroker.default-topic=IBM-SYSAUTO-SDF-&SYSPLEX.
```

On PLEX1, this setting resolves to *ing.infobroker.default-topic=IBM-SYSAUTO-SDF-PLEX1*.

On PLEX2, this setting resolves to *ing.infobroker.default-topic=IBM-SYSAUTO-SDF-PLEX2*.

Next step

“Step 22E: Customize Properties of the System Automation Info Broker” on page 116

Step 22E: Customize Properties of the System Automation Info Broker

This Java process of the System Automation Info Broker is started by the System Automation Info Broker components residing in NetView on request. This is being controlled with the configuration you have done in the previous steps. The Java process (also called the "System Automation Info Broker Publisher") can be configured with help of the contents of the following sample files that can be found in `/usr/lpp/ing/infobroker/config`.

Sample Configuration File	Description
ing.infobroker.environment	This environment file is needed to set up the USS environment for the System Automation Info Broker Publisher - the Java process forwards the messages to your Kafka server. See “Step 22D: Set up the JCL Procedure and the Environment File” on page 115. System symbols are supported in this file.
ing.infobroker.properties	General runtime parameters of the System Automation Info Broker Publisher Process. Configure here where your Kafka server is located and on which port it is listening. System symbols are supported in this file.
ing.infobroker.security.properties	Optional Security configurations in order to secure the connection path to your Kafka server. System symbols are supported in this file.
ing.infobroker.logging.xml	This XML configures the tracing done by the System Automation Info Broker Publisher Process.

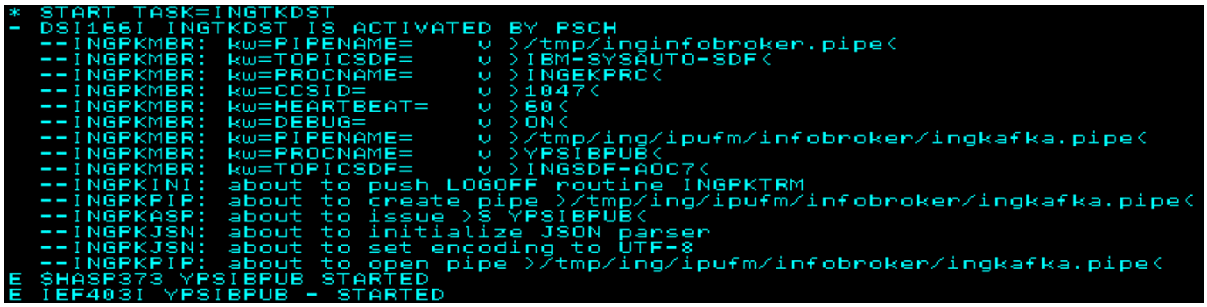
Procedure

1. Copy the sample configuration files from the directory
   `/usr/lpp/ing/infobroker/config`
   to your custom directory created in “Step 22A: Set up the USS Directory Structure” on page 113, for example,
   `/etc/ing/infobroker/config`
2. Customize the properties files to your needs as instructed by the descriptions within that sample files. If you want to use system symbols in these properties files, see “System Symbol Support for System Automation Info Broker (OA62518)” on page 115.

Step 22F: Start and Stop the System Automation Info Broker

Start the System Automation Info Broker

The System Automation Info Broker is automatically started by the NetView initialization. You can check in the Netlog to see whether it's activated.



If you have stopped System Automation Info Broker or you have not set to automatically start it by the NetView initialization (that is, **TASK.INGTKDST.INIT** is not set to Y in the NetView style sheet member), you can manually start it with the following command.

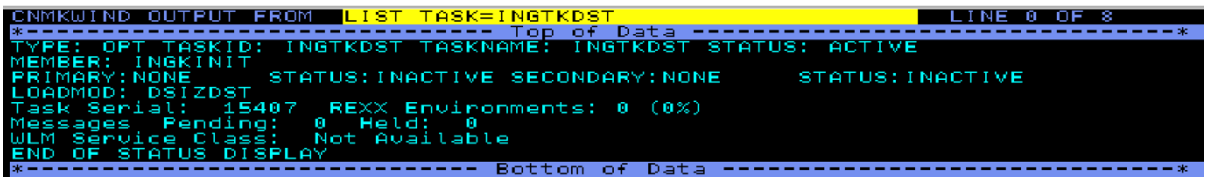
```
START TASK=INGTKDST
```

Verify whether System Automation Info Broker is running

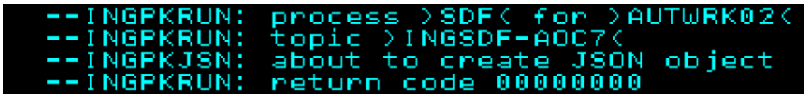
To verify whether System Automation Info Broker is running, you can use either of the following ways:

- Issue the following command on NetView to check the task status. If the command output data contains `INGTKDST STATUS: ACTIVE`, it indicates that the Info Broker is running.

  ```
  LIST TASK=INGTKDST
  ```

- If debug is enabled (**DEBUG** = NO) in the INGKINIT member, you can check the Netlog. If messages are sent, it indicates that the Info Broker is running.

  Or if the hear beat is still working, it indicates that the Info Broker is running.

Stop the System Automation Info Broker

To stop System Automation Info Broker, you can use the following three staggered stop commands. Use the force or even immediate shutdown commands, if normal shutdown is not working.

- Normal shutdown. It can take up to 30s to shut down the Info Broker.

  ```
  STOP TASK=INGTKDST
  ```

- Force shutdown.

  ```
  STOP FORCE=INGTKDST
  ```

- Immediate shutdown.

  ```
  STOP IMMED=INGTKDST
  ```

Step 23: Check for Required IPL

SysOps	ProcOps
√	√

An IPL is only required if:

- In “Step 4A: Update IEAAPFxx” on page 73, you used the IEAAPF*xx* member to define authorized libraries to the APF
- In “Step 4D: Update LPALSTxx” on page 74 you decided ***not*** to use the solution to dynamically add the modules to the LPALST

- In “Step 4E: Update LNKLSTxx” on page 74 you updated LNKLST and you decided ***not*** to use the solution to dynamically add the modules to the LNKLST
- “Step 4G: Update IEFSSNxx” on page 75 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 24: Automate System Operations Startup

About this task

SysOps	ProcOps
√	√

Sample: INGECOM

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.

Make the described changes to the following SYS1.PARMLIB data set members:

Sample 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 CNMSJ010,JOBNAME=SYSVSSI,SUB=MSTR'
   COM='S INGENVSA,JOBNAME=SYSVAPPL,SUB=MSTR'
   ```

   Note:

   CNMSJ010 is the name of the sample that is provided by NetView that you copied in “Step 5: Configure SYS1.PROCLIB Members” on page 76.

   INGENVSA is the name of the sample that is provided by SA z/OS that you copied in “Step 5: Configure SYS1.PROCLIB Members” on page 76.
3. Adapt the NetView Application and NetView Subsystem Interface jobname to agree with the four-character prefix defined in the IEFSSNxx member, which is described in “Step 4G: Update IEFSSNxx” on page 75. For example, if the name of the NetView Application jobname is SYSVxx, SYSV must be specified in the IEFSSnxx member as the character prefix.

Sample 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

About this task

Note: The system that the automation manager should be started on 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 109.

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:

```
S INGEAMSA,JOBNAME=AM,SUB=MSTR
```

You can find the sample startup procedure called INGEAMSA in the SINGSAMP sample library.

Step 25: Verify Automatic System Operations Startup

About this task

SysOps	ProcOps
*	

After you have installed the host components of SA z/OS, it is recommended that you perform the following steps for verification purposes:

Procedure

1. Perform an IPL, if you have not done this according to "Step 23: Check for Required IPL" on page 117. Then start SA z/OS.

   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 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 Z System Automation 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 Z System Automation 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 26: Configure USS Automation

SysOps	ProcOps
*	

Step 26A: Securing USS Resources

When setting up USS automation then SAF related actions are required. Therefore the user IDs:

- must have an OMVS segment
- must be permitted to the appropriate SAF profiles.

Refer to the chapter Chapter 11, “Security and Authorization,” on page 151 to generate a security definition member INGESAF by using the configuration assistant.

Each user ID definition in INGESAF contains an OMVS sgement (ADDUSER/ALTUSER). The section *Set OMVS Security* within this member lists the USS SAF profiles.

Step 26B: Preparing for USS Automation

About this task

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 27: Configure and Run the System Automation Data Store

SysOps	ProcOps
*	

Prerequisites

Ensure that the following prerequisites are met before you can configure and run the System Automation data store.

- Java™ Runtime Environment (JRE) 1.8 or higher installed on z/OS. The 32-bit Java version is required.
- SSI address space with PPI function of IBM Z NetView.
- z/OS UNIX System Services (USS) available with mounted USS file system for use by the System Automation data store.

Procedure

- "Step 27A: Set up the USS Directory Structure" on page 121
- "Step 27B: Set up the JCL Procedure and the Environment File" on page 121
- "Step 27C: Set up the Data Store Properties File" on page 122
- "Step 27D: Start and Stop the System Automation Data Store" on page 123

Step 27A: Set up the USS Directory Structure

About this task

After SMP/E installation of System Automation, there is a USS directory structure for System Automation data store.

- `/usr/lpp/ing/datastore/lib`

  This is the read-only directory that contains the executables.
- `/usr/lpp/ing/datastore/config`

  This directory contains the configuration files that you need to copy into your custom directory.

Procedure

Create your own working directory to your needs. Assume the custom directory is `/var/ing/datastore` and you have created the following subdirectories.

- `/var/ing/datastore`
- `/var/ing/datastore/data`

Step 27B: Set up the JCL Procedure and the Environment File

The sample procedure ING.SINGSAMP(INGDBJCL) starts the Java application of System Automation Data Store.

The //STDENV statement points to an environment file that resides in your custom directory. This environment file is needed to set up the USS environment for the System Automation Data Store.

Procedure

1. Modify the **`CONFIG`** parameter in the sample procedure to the custom directory of your choice.
2. Modify the VERSION parameter in the sample procedure to reflect the Java version installed on your system (for example 17 or 21).
3. Copy the sample environment file from the directory `/usr/lpp/ing/datastore/config/ing.datastore.environment` to your custom directory `/var/ing/datastore/ing.datastore.environment`
4. Customize the variables in the environment file as instructed by the descriptions within that sample file.

System Symbol Support for System Automation Data Store (OA62518)

This feature resolves system symbols in the configuration variables at Data Store startup. It enables you to have a generic configuration file that acts as a "single source of truth" and can be shared across systems and sysplexes. The specific configuration for each system can then be controlled via system symbols.

By default, this feature is disabled. To enable this feature, navigate to the file '`ing.datastore.environment`' and set the following property to true.

```
# Enable system symbol support [true/false].
ENABLE_SYSTEM_SYMBOL_SUPPORT=true
```

After the feature is enabled, you can use system symbols in both '`ing.datastore.environment`' and '`ing.datastore.properties`' files.

Example 1: ing.datastore.environment

```
DATASTORE_CONFIG_HOME=/var/&SYSNAME./ing/datastore
```

On SYS1, this setting resolves to *DATASTORE_CONFIG_HOME=/var/SYS1/ing/datastore*.

On SYS2, this setting resolves to *DATASTORE_CONFIG_HOME=/var/SYS2/ing/datastore*.

Example 2: ing.datastore.properties

```
ing.datastore.db.path=/var/&SYSPLEX./ing/datastore/data
```

On PLEX1, this setting resolves to *ing.datastore.db.path=/var/PLEX1/ing/datastore/data*.

On PLEX2, this setting resolves to *ing.datastore.db.path=/var/PLEX2/ing/datastore/data*.

Step 27C: Set up the Data Store Properties File

Procedure

1. Copy the properties file from the directory `/usr/lpp/ing/datastore/config/ing.datastore.properties` to your custom directory `/var/ing/datastore/ing.datastore.properties`
2. Customize the root directory `/var/ing/datastore` to your needs. The root directory must be the same as specified in the parameter *DATASTORE_CONFIG_HOME* in file `ing.datastore.environment`.
3. Customize the PORT, KEYSTORE and TRUSTSTORE to the following values:

Property	Value	Description
ing.service.port	*<PORT>*, for example *8181*	The port on which the System Automation Data Store is listening.
ing.security.tls.keystore / *ing.security.tls.truststore*	*safkeyring://<USERID>/<KEYRING_NAME>* OR */path/to/keystore.jks*	The URL that specifies the TSO user ID (owner of the key ring) and the name of the RACF or ICSF key ring to use as keystore or truststore.
ing.security.tls.keystore.type / *ing.security.tls.truststore.type*	*JCERACFKS*(for RACF) or *JKS*(for Java Keystore)	The RACF or JKS keystore type.

Property	Value	Description
ing.security.tls.keystore.password / *ing.security.tls.truststore.password*	*<your secret>* OR *password*	Password for your Keystore/ Truststore. RACF key rings are not password protected. Instead, the necessary permissions are checked for the user ID specified in the keystore/truststore properties. Since an empty value is not allowed, the value must be *'password'*.
ing.security.tls.key.alias	*<LABEL>*, for example *tomcat*	The label (alias) used to connect the own certificate with the RACF key ring. Not required if the key ring is used as truststore.

4. Optional: If you want to have the System Automation Data Store to write debug statements into the log directory, set the debug option *ing.datastore.logging.level* from INFO to DEBUG.
5. Optional: If you want to use system symbols in this properties file, see System Symbol Support for System Automation Data Store.

Step 27D: Start and Stop the System Automation Data Store

The System Automation data store runs in a Java virtual machine in a USS process. To start the data store, run a started task by using the sample JCL procedure ING.SINGSAMP(INGDBJCL). To stop it, use the z/OS stop command.

You also need to make the System Automation data store highly available. To do so, define the System Automation data store as an APL with category INGDB in the policy. The data store must run once per SAplex. For that purpose, the new application category INGDB is introduced. Instead of defining the data store policy yourself, you can also import the data store sample policy from *BASE via component **System automation data store**.

Step 28: Configure Db2 as an alternative database of dynamic resources

You can choose to store dynamic resources either in System Automation data store or IBM Db2 for z/OS (the latter alternative is enabled with OA63123). Complete this configuration if you choose IBM Db2 for z/OS as the database.

Before you begin

IBM Db2 for z/OS V12 or later must be installed. If not, install it by following IBM Db2 for z/OS V12 documentation.

About this task

System Automation delivers the Data Definition Language (DDL) files that are used to configure Db2 as data set members. You can either use SPUFI to run the DDL files or issue the statements from any client tool, for example IBM Data Studio or DBVisualizer.

You might need to add security profile ssid.**BATCH** to resource class for Db2 **DSNR**. This is required because System Automation accesses the database through DSNREXX.

The following DDL files are provided as sample members:

Member name	Short description	Description
INGD2PFC	CREATE/GRANT-statements for Stored Procedures & Functions	Sample DDL file to create all stored procedures in Db2 to work with INGDYN and GRANT EXECUTE privileges for all INGDYN-related stored procedures in Db2.
INGD2PFD	DROP/REVOKE-statements for Stored Procedures & Functions	Sample DDL file to drop all INGDYN-related stored procedures in Db2, and revoke privileges on functions and stored procedures.
INGD2TVC	CREATE/GRANT-statements for Tables & Views	Sample DDL file to create all tables and views for a given SAPlex in Db2 to work with INGDYN, and GRANT ALL privileges to access the data with INGDYN for a given SAPlex.
INGD2TVD	DROP/REVOKE-statements for Tables & Views	Sample DDL file to drop all INGDYN-related tables and views for a given SAPlex, and revoke privileges on one or more tables or views.

Procedure

On the system where Db2 is running, complete these steps.

1. Install the OA63123 APAR.
2. Code page IBM 1047 is used for the DDL files that are delivered by System Automation. If you use a different code page, manually convert the data set members to the code page that you are using in your Db2 subsystem.
3. Run the DDL files to create Db2 objects. You can either use SPUFI (SQL processor using file input) facility to run the DDL files or issue the statements from any client tool like for example, IBM Data Studio or DBVisualizer.

   If you want to execute the SQL with SPUFI, you need to have six members in the data set as shown in the table above.

   - Scenario 1: first installation and configuration

     a. If you want to configure an SAPlex to work with INGDYN and Db2 for the very first time, you first need to set up the "Stored Procedures" and "User-defined Functions" in Db2 that INGDYN requires. To do this, you need to adapt and run the sample members INGD2PFC and INGD2PFP. The "Stored Procedures" and "User-defined Functions" must be placed in a special schema for example IBMING to ensure that the "Stored Procedures" and "User-defined Functions" only exist once in the Db2 cluster.

     b. After the "Stored Procedures" and "User-defined Functions" are in place, you can next adapt and run the sample members INGD2TVC and INGD2TVP to create the necessary tables and views and grant the needed privileges to access the data.
     Here you should create and use a separate schema for each SAPlex from which you want to use INGDYN with Db2 as backend.

   - Scenario 2: Add a new SAplex

     If you want to use INGDYN with Db2 on a new SAPlex, you must first drop all the existing DDL and then run other members.

   For more information, see Executing SQL by using SPUFI in IBM Db2 for z/OS documentation.
4. Recycle NetView to refresh the PTF changes.
5. If you are going to use Db2 as the INGDYN backend on this system where Db2 is running, specify the advanced automation variable AOF_DYNRES_DB_EXTINFO in your NetView stylesheets. INGDYN

calls the Db2 backend if this variable exists and has correct syntax. For more information about this variable, see *IBM Z System Automation Customizing and Programming*.

On the other systems where to use Db2 as the database of dynamic resources, complete these steps.

1. Install the OA63123 APAR.
2. Specify the advanced automation variable AOF_DYNRES_DB_EXTINFO in your NetView stylesheets.

Step 29: Configure and Run the System Automation Operations REST Server

SysOps	ProcOps
*	

The Operations REST Server is an optional component of IBM Z System Automation. It provides an application programming interface (API) and allows the programmatic exploitation of resources that are automated by System Automation. Developers of other programs or products that allow to exploit REST interfaces can use this component to add System Automation operations as part of their workflow.

You need to decide what option you want to follow per SAplex, and then deploy and configure the Operations REST Server as described in the following steps.

[OPT_EMBEDDED_WEBSERVER]

The Operations REST Server runs as a standalone version with its own embedded web server. This is the quickest option to run the Operations REST Server.

Ensure the following prerequisites are met before you choose this option:

- IBM Z NetView 6.3.0 or later installed.
- Java Runtime Environment (JRE) 17 or higher installed on z/OS.

  Starting from APAR OA65704, Java 17 or higher is required. If you have previously used Operations REST Server with an earlier Java version, follow the migration steps in Step E: Migrate to Java 17 / Java 21 (OA65704).
- SSI address space with PPI function of IBM Z NetView.
- z/OS UNIX System Services (USS) available with mounted USS file system.

[OPT_LIBERTY_DEPLOYED]

The Operations REST Server runs as a deployed web application within IBM WebSphere Liberty.

Ensure the following prerequisites are met before you choose this option:

- IBM Z NetView 6.3.0 or later installed.
- IBM WebSphere Liberty installed. Either WebSphere Liberty feature `servlet-5.0` or `servlet-6.0` is enabled.
- Java Runtime Environment (JRE) 17 or higher installed on z/OS.

  Starting from APAR OA65704, Java 17 or higher is required. If you have previously used Operations REST Server with an earlier Java version, follow the migration steps in Step E: Migrate to Java 17 / Java 21 (OA65704).
- SSI address space with PPI function of IBM Z NetView.
- z/OS UNIX System Services (USS) available with mounted USS file system.

Procedure

Step 29A: Set up the USS Directory Structure

After SMP/E installation of System Automation, the following USS directories are available for the System Automation Operations REST Server.

USS Directory	OPT_EMBEDDED_WEBSERVER	OPT_LIBERTY_DEPLOYED
/usr/lpp/ing/restsrvr/lib	This is the read-only directory that contains the executables.	This directory contains the Web Application Archive (WAR) file that can be used to deploy the server to WebSphere Liberty.
/usr/lpp/ing/restsrvr/config	This directory contains the sample configuration files that you need to copy into your custom directory that holds the configuration data. Adjust the configuration files to your needs.	N/A
/usr/lpp/ing/restsrvr/samples	This directory contains miscellaneous sample files, for example, configurations that can be used to integrate the Operations REST Server into other products, such as the Zowe offering.	

Customize your configuration directory

This step is needed only for the OPT_EMBEDDED_WEBSERVER option.

Create your own configuration directory to your needs. Assume that you put configuration data to the `/etc` directory, you should create the following directory:

`/etc/ing/restsrvr`

Ensure that this directory is readable by the process running the Operations REST server.

Step 29B(I): Configurations for the OPT_EMBEDDED_WEBSERVER Option

This topic describes the configuration steps for the OPT_EMBEDDED_WEBSERVER option. If you're using the OPT_LIBERTY_DEPLOYED option, see “Step 29B(II): Configurations for the OPT_LIBERTY_DEPLOYED Option” on page 131.

Set up the JCL procedure

The sample procedure ING.SINGSAMP(INGROJCL) starts the Java application of System Automation Operations REST Server.

You need to modify the CONFIG parameter in the sample procedure to the custom directory of your choice. For example, '`/etc/ing/restsrvr/config`'.

```
//INGROSRV PROC CONFIG='/etc/ing/restsrvr/config',
//   JAVACLS='org.springframework.boot.loader.launcher.JarLauncher',
//   LOGLVL='+I',
//   LEPARM=''
...
```

Set up the environment file, properties file, security, tracing and logging

The following sample configuration files can be found in `/usr/lpp/ing/restsrvr/config`. You need to customize these file before you can start the Operations REST Server.

Sample Configuration File	Description
ing.operations.environment	This environment file is needed to set up the USS environment for the Operations REST Server. System symbols are supported in this file.
ing.operations.properties	This file is used to control various parameters of the Operations REST Server and the integration with other products, such as Zowe. System symbols are supported in this file.
ing.operations.security.properties	This file is used to set up network security. If you want to set the rate limiter to control the maximum number of calls to the Operations REST API, see Rate Limiter. System symbols are supported in this file.
ing.operations.logging.xml	This file controls how the Operations REST Server writes logs and traces.

1. Copy the sample configuration files from the directory `/usr/lpp/ing/restsrvrv/config` to your custom directory created in step 29A, for example, `/etc/ing/restsrvr/config`
2. Customize the configuration files to your needs as instructed by the descriptions within the sample files. Here are some considerations that you need to be aware during your customization:
   - To use system symbols in these configuration files, see System Symbol Support for System Automation Info Broker.
   - To enable RACF or ICSF key kings, see Configure the properties for using RACF or ICSF key rings.
   - To set the maximum Operations REST API calls for security reasons, see Rate Limiter.

System Symbol Support for Operations REST Server (OA62518)

This feature resolves system symbols in the configuration variables at startup. This enables the user to have a generic configuration file that acts as a "single source of truth" and can be shared across systems and sysplexes. The specific configuration for each system can then be controlled via system symbols.

By default, this feature is disabled. To enable this feature, navigate to the file 'ing.operations.environment' and set the following property to true.

```
# Enable system symbol support [true/false].
ENABLE_SYSTEM_SYMBOL_SUPPORT=true
```

After the feature is enabled, you can use system symbols in the '*ing.operations.environment*', '*ing.operations.properties*', and *ing.operations.security.properties* files.

Example 1, `ing.infobroker.environment`:

```
REST_API_CONFIG_HOME=/etc/&SYSNAME./ing/restsrvr
```

On SYS1, this setting resolves to *REST_API_CONFIG_HOME=/etc/SYS1/ing/restsrvr*.

On SYS2, this setting resolves to *REST_API_CONFIG_HOME=/etc/SYS2/ing/restsrvr*.

Example 2, `ing.operations.properties`:

```
ing.service.port=&SAROPORT.
```

On PLEX1, this setting resolves to *ing.service.port=8443*.

On PLEX2, this setting resolves to *ing.service.port=9443*.

Configure the properties for using RACF or ICSF key rings (OA62518)

System Automation Operations REST Server can use RACF or ICSF (Integrated Cryptographic Service Facility) key rings to manage certificates if configured accordingly for the OPT_EMBEDDED_WEBSERVER option.

A single RACF or ICSF key ring can be used as keystore and truststore in parallel, holding the required certificate for the embedded web server to provide TLS/HTTPS and the necessary CA certificates for validation, for example, if Zowe is used for JWT token authorization.

If desired, two RACF or ICSF key rings can be used for that purpose as well. It is also possible to use two different keystore and truststore types in parallel. For example, store the own certificate for TLS/HTTPS in a RACF or ICSF key ring and manage the trusted certificates in a PKCS12 file in USS.

To use a RACF or ICSF key ring, complete the following configurations:

1. Navigate to the `ing.operations.environment` configuration file. Ensure that the *ENABLE_KEYRING_SUPPORT* property is set to true and use appropriate *IBM_CRYPTO_PROVIDER* value. Starting from OA62518, RACF key ring is enabled by default.
   - For RACF key ring, use `IBM_CRYPTO_PROVIDER="com.ibm.crypto.provider"`
   - For ICSF key ring, use `IBM_CRYPTO_PROVIDER="com.ibm.crypto.hdwrCCA.provider"`
2. Navigate to the `ing.operations.security.properties` configuration file, set the keystore, truststore, or both properties to the following values:

Table 18. Properties for using RACF or ICSF key rings

Properties	Value	Description
ing.security.tls.keystore / ing.security.tls.truststore	safkeyring:////<*USERID*>/ <*KEYRING_NAME*>	The URL that specifies the TSO user ID (owner of the key ring) and the name of the RACF or ICSF key ring to use as keystore or truststore. Note that the slashes after the protocol type *safkeyring* needs to be escaped, requiring a total of four slashes.
ing.security.tls.keystore.type / ing.security.tls.truststore.type	To use a RACF key ring, set the value to JCERACFKS. To use an ICSF key ring, set the value to JCECCARACFKS.	The RACF or ICSF keystore type.

Table 18. Properties for using RACF or ICSF key rings (continued)		
ing.security.tls.keystore.password / ing.security.tls.truststore.password	password	RACF or ICSF key rings are not password protected. Instead, the necessary RACF or ICSF permissions are checked for the user ID specified in these properties: • ing.security.tls.keystore • ing.security.tls.truststore Since an empty value is not allowed for this property, the value **must** be 'password'.
ing.security.tls.key.alias	<*LABEL*>, for example, tomcat	The label (alias) that is used to connect the own certificate with the RACF or ICSF key ring. Not required if a RACF or ICSF key ring is used as truststore.

For details of how to manage certificates with RACF, see Using RACF with Encryption Facility in z/OS documentation. For more information about ICSF, see Get Started with ICSF in z/OS documentation.

Configure the properties for using RACF or ICSF key rings (OA65704)

To use a RACF or ICSF key ring, complete the following configurations:

1. Navigate to the `ing.operations.environment` configuration file. Ensure that the *ENABLE_KEYRING_SUPPORT* property is set to true and use appropriate *IBM_CRYPTO_PROVIDER* value. Starting from OA62518, RACF key ring is enabled by default.
   - For RACF key ring, use `IBM_CRYPTO_PROVIDER="com.ibm.crypto.zsecurity.provider"`
   - For ICSF key ring, use `IBM_CRYPTO_PROVIDER="com.ibm.crypto.hdwrCCA.provider"`
2. Navigate to the `ing.operations.security.properties` configuration file, set the keystore, truststore, or both properties to the following values:

Table 19. Properties for using RACF or ICSF key rings		
Properties	Value	Description
ing.security.tls.keystore / ing.security.tls.truststore	safkeyring://<*USERID*>/<*KEYRING_NAME* safkeyring://<*USERID*>/<*KEYRING_NAME*>	The URL that specifies the TSO user ID (owner of the key ring) and the name of the RACF or ICSF key ring to use as keystore or truststore.
ing.security.tls.keystore.type / ing.security.tls.truststore.type	To use a RACF key ring, set the value to JCERACFKS. To use an ICSF key ring, set the value to JCECCARACFKS.	The RACF or ICSF keystore type.

Table 19. Properties for using RACF or ICSF key rings (continued)		
ing.security.tls.keystore.password / ing.security.tls.truststore.password	password	RACF or ICSF key rings are not password protected. Instead, the necessary RACF or ICSF permissions are checked for the user ID specified in these properties: • ing.security.tls.keystore • ing.security.tls.truststore Since an empty value is not allowed for this property, the value **must** be 'password'.
ing.security.tls.key.alias	<*LABEL*>, for example, tomcat	The label (alias) that is used to connect the own certificate with the RACF or ICSF key ring. Not required if a RACF or ICSF key ring is used as truststore.

Rate Limiter (OA62518)

A rate limit is the maximum number of calls you want to allow in a particular time interval. Setting rate limits enables you to manage the network traffic for this Operations REST API. With the rate limiter, you can specify the number of calls to be accepted within a defined time period. For example:

- 5 calls per 10 second
- 100 calls per minute
- 200 calls per hour

Rate limits are hard. It means if a call exceeds the limit, then the call is aborted and an error is returned. When the rate limit is reached, no more calls are accepted from that user until the beginning of the next time period. For example, you might want to permit a total of 1000 calls per hour (rate limit). If a user makes 1000 calls in the first 10 minutes, they cannot complete any more calls until the hour has expired.

Rate limit is enabled by default for each unique IP address or each authenticated user ID. The rate limit for IP addresses will be applied for every API call before the user authentication happens. The rate limit for user IDs will be applied after a successful authentication. By default, rate limiter allows a maximum of 100 requests within 1 minute for each unique IP address and a maximum of 60 requests within 1 minute for each authenticated user ID.

Note: Since this feature is security-related, it is an opt-out approach. Keep in mind that rate limiter is a breaking change as it can cause issues for client applications that do not handle the appropriate HTTP error code.

To configure rate limit settings, modify the rate-limit-related properties as described in the following table or in the `ing.operations.security.properties` file.

Properties	**Description**
ing.security.rate-limit.enabled	Enable or disable the security feature to limit the number of API requests from a single IP address or a user ID.

Properties	Description
ing.security.rate-limit.<type>.enabled	Enable or disable the security feature to limit the number of API requests for the specific type. Valid types are: • ip-address: rate limit for unique IP addresses. • userid: rate limit for each authenticated user ID. **Note:** The rate limit for IP addresses will be applied for every API call before the user authentication happens. The rate limit for user IDs will be applied after a successful authentication.
ing.security.rate-limit.<type>.requests	Maximum number of requests that should be allowed within the configured time period from a unique IP address or a unique user ID.
ing.security.rate-limit.<type>.time	Amount of time after which the rate limit should be completely reset. By default, this configuration allows a maximum of 100 requests within 1 minute for each unique IP address.
ing.security.rate-limit.ip-address.unit	Time unit allowed for the rate limit. Valid units are: • seconds • minutes • hours • days

Step 29B(II): Configurations for the OPT_LIBERTY_DEPLOYED Option

This topic describes the configuration steps for the OPT_LIBERTY_DEPLOYED option. If you're using the OPT_EMBEDDED_WEBSERVER option, see “Step 29B(I): Configurations for the OPT_EMBEDDED_WEBSERVER Option” on page 126.

Deploy the Operations REST Server

You need to deploy the Operations REST Server "war" file to your WebSphere Liberty environment, adapt the JCL that starts the WebSphere server instance, and customize the `server.xml` and `server.env` files.

Notice: The REST API requires either ‘servlet-5.0’ or ‘servlet-6.0’ WLP features.

Procedure

1. Choose an existing or create a new liberty server instance to host the System Automation Operations REST Server that is found as a "war" file in the directory `/usr/lpp/ing/restsrvr/lib`. Use your standard procedure to deploy this "war" file as "APP". For example, copy it into the `/apps` directory of the liberty server instance directory that you have chosen to host the Operations REST Server.
2. Modify the **`CONFIG`** parameter in the start procedure of your WebSphere Liberty server that you have chosen to deploy the Operations REST Server.
3. Adapt the files "`server.xml`" and "`server.env`" that have been created by the liberty-owned command to create a new liberty server instance.

   In the server.xml file, define the name of your System Automation Operations REST Server running as a liberty APP, the context-root for the app and port(s).

Here is an example excerpt of `server.xml` . Choose the values that are suitable for your environment.

```
...
<httpEndpoint id="defaultHttpEndpoint" host="*" httpPort="9081" httpsPort="9444"/>
...
<application context-root="/ibm/sa" type="war" id="SAREST" location="ingoperations.war"
name="SAREST" />
...
```

The configuration of the System Automation Operations REST Server is done with the environment variables which have to be added to the `server.env` file. For example, you can configure tracing and logging. See Table 20 on page 132 for all the properties and descriptions.

4. Make sure that your <NETVIEW_V63_HOME>/`restsrvr/bin` is added to the **PATH** environment of the user that starts the Operations REST Server.

Table 20. Properties in the server.env file

Properties	Description	Default
SPRINGDOC_SWAGGER_UI_ENABLED	Swagger UI	true
ING_OPERATIONS_LOG_LEVEL	Sets the log level for the application.	INFO
ING_OPERATIONS_{LOG,TRACE}_FILE_PATH	Sets the path where the {log,trace} files will be written.	/var/log/ing/restsrvr
ING_OPERATIONS_{LOG,TRACE}_FILE_NAME	Sets the name of the {log,trace} file.	{ing.operations,ing.operations.trace}
ING_OPERATIONS_{LOG,TRACE}_FILE_ARCHIVE_PATTERN	The file name pattern for archived {log,trace} files.	/var/log/ing/restsrvr/%d{yyyy-MM}/{ing.operations,ing.operations.trace}-%d{yyyy-MM-dd}-%i
ING_OPERATIONS_{LOG,TRACE}_FILE_DELETE_AFTER	Specify the number of days for which {log,trace} files are kept. If a log file is older than the specified period, it is automatically deleted.	{730,10} (in days)
ING_OPERATIONS_{LOG,TRACE}_FILE_MAX_PER_DAY	Specify how many {log,trace} files may be created per day before old log files are overwritten.	{10,100}
ING_OPERATIONS_{LOG,TRACE}_FILE_MAX_SIZE	Specify the maximum size of the {log,trace} file (in megabytes). If the maximum size is exceeded, a new {log,trace} file is automatically created.	20 (in MB)
ING_OPERATIONS_ZOWE_URLS_QUERY	Specify the URL where the Zowe API Mediation Layer validates the JWT token.	${zowe.urls.apiml}/api/v1/gateway/auth/query

Configurations for using RACF key rings

The System Automation Operations REST Server can use RACF key rings to manage certificates if configured accordingly for the OPT_LIBERTY_DEPLOYED option. Integrated Cryptographic Service Facility (ICSF) key rings are not supported.

For details of the configuration steps, see Configuring SAF certificates and keyrings for TLS on the z/OS operating system in WebSphere Application Server for z/OS Liberty documentation.

Step 29C: Enable the NetView PPI

This step is required for both OPT_EMBEDDED_WEBSERVER and OPT_LIBERTY_DEPLOYED options.

The System Automation Operations REST Server shares the PPI communication to NetView with a component delivered by NetView 6.3. It is not required that the NetView REST server is started as prerequisite to run System Automation Operations REST Server. However, the PPI communication has to be enabled.

Ensure that the command receiver auto task has been started. For example, you can enable it in C*xx*STGEN in the following way:

```
function.autotask.RESTOP = AUTOREST //Command receiver auto task
AUTOTASK.?RESTOP.Console = *NONE*
AUTOTASK.?RESTOP.InitCmd = CMDSERV NAME=EJNREST,AUTHSNDR=NO
```

Step 29D: Start and Stop the System Automation Operations REST Server

Start the Operations REST Server

- OPT_EMBEDDED_WEBSERVER

  The System Automation Operations REST Server runs in a Java™ virtual machine in a USS process. To start this server, run a started task by using the sample JCL procedure ING.SINGSAMP(INGROJCL).
- OPT_LIBERTY_DEPLOYED

  Normally, the System Automation Operations REST Server is automatically started as an APP when you start the WebSphere Liberty server instance. It is possible that you configured another behavior in the liberty profile. Consult your WebSphere Liberty expert to clarify the start procedure in your environment.

Messages are generated and written to the Netlog whenever the System Automation Operations REST server is starting. ING420I and ING421I indicate a successful start.

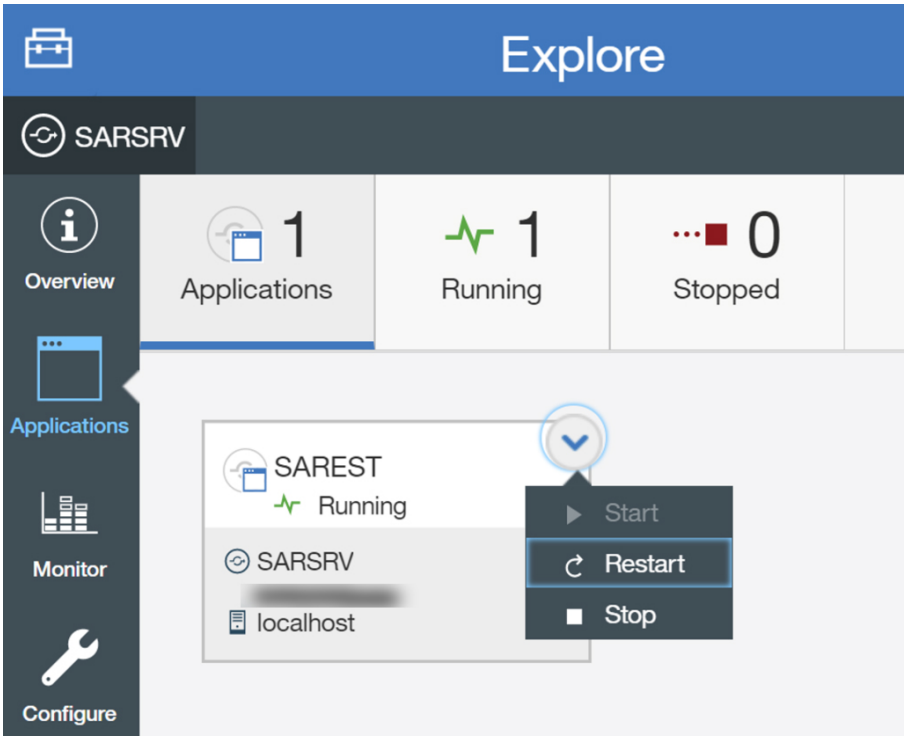
Stop the Operations REST Server

- OPT_EMBEDDED_WEBSERVER

  The System Automation Operations REST Server runs in a Java™ virtual machine in a USS process. To stop it, use the z/OS® stop command.
- OPT_LIBERTY_DEPLOYED

  The System Automation Operations REST Server is stopped when you stop the WebSphere Liberty server instance.

  It is possible to stop and restart individual applications in the WebSphere Admin Console. For more information about how to do it, see the description of you local WebSphere Liberty Deployment.



Messages are generated and written to the Netlog whenever the System Automation Operations REST Server is stopping. ING422I and ING423I are written during stop processing.

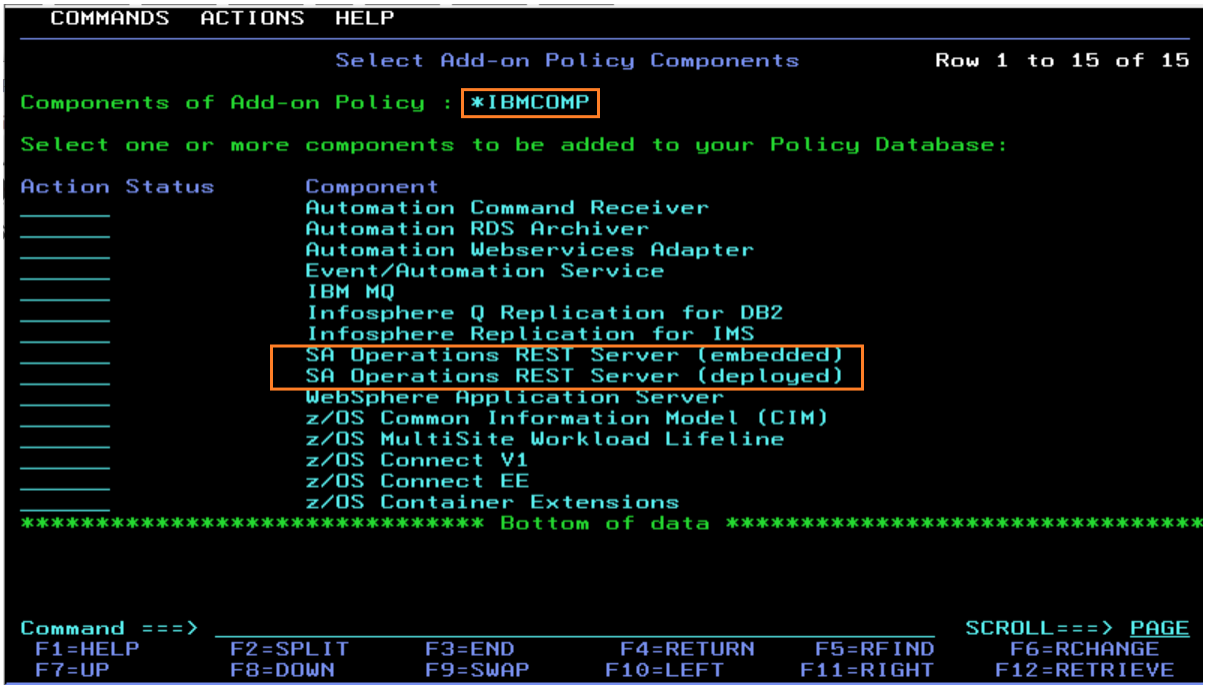
Automate start and stop of the Operations REST Server

To automate the start and stop of the Operations REST Server, you can use the sample policies that are provided by System Automation in the Customization Dialog.

Two sample policies are available under the *IBMCOMP add-on policy in the Customization Dialog (**Data Management** > **Import from Add-on**).

- One is for OPT_EMBEDDED_WEBSERVER (the Operations REST Server runs as a stand-alone version with its own embedded Tomcat web server).
- The other is for OPT_LIBERTY_DEPLOYED (you deploy the Operations REST Server in your WebSphere Liberty environment).

You can add this component to your policy as an embedded or WebSphere deployed solution and adapt the sample policies to your needs.



Step 30: Configure the Policy Services Provider

SysOps	ProcOps
*	

Policy Services Provider is an optional component of Z System Automation. If you want to generate the policy database report in JSON format, you have to configure this component.

Prerequisite

Ensure that the following prerequisites are met before you configure and run the Policy Services Provider.

- Java Runtime Environment (JRE) 17 or higher installed on z/OS.
- z/OS Client Web Enablement Toolkit is available.

Steps

- "Step 30A: Set up the USS Directory Structure" on page 135
- "Step 30B: Set up configuration files" on page 136
- "Step 30D: Secure the Policy Services Provider" on page 136
- "Step 30E: Start and Stop the Policy Services Provider" on page 137

Step 30A: Set up the USS Directory Structure

After the SMP/E installation of System Automation, the following USS directories are available as part of the Policy Services Provider structure.

USS directory	Description
`/usr/lpp/ing/policyservices`	This is the read-only directory that contains the Policy Services Provider component files.

USS directory	Description
/usr/lpp/ing/policyservices/bin	This is the read-only directory that contains Policy Services Provider application versions.
/usr/lpp/ing/policyservices/config	This is the read-only directory that contains the configuration file sample.
/usr/lpp/ing/policyservices/lib	This is the read-only directory that contains the executable modules.

Customize your configuration directory

Create your own site-specific configuration directory to your needs. Assume that the working directory is /etc/ing/policyservices, and you created the following sub-directories.

USS directory	Description
/etc/ing/policyservices	This directory hosts the configuration file of the SA Policy Services Provider.

Next step

"Step 30B: Set up configuration files" on page 136

Step 30B: Set up configuration files

This topic describes the configuration steps for the Policy Services Provider (PSP).

The following sample configuration files can be found in

/usr/lpp/ing/policyservices/config

You need to customize these files before you can start the SA Customization REST Server.

Sample configuration file	Description
ing.policyservices.properties	This file is used to control various parameters of the SA Customization REST Server . This is the main application configuration file for the SA Customization REST Server, that includes hostname, port, logging settings, ISPF bridge, and OpenAPI support.
ing.policyservices.security.properties	This file is used to set up network security. This is for Security configuration for the SA Customization REST Server, that includes TLS enablement, keystore, and truststore settings.

1. Copy the sample configuration files from the directory.

   /usr/lpp/ing/policyservices/config to your custom directory for example, /etc/ing/policyservices
2. Customize the configuration files to your needs as instructed by the descriptions within the sample files.

Step 30D: Secure the Policy Services Provider

The SA Customization API supports two authentication methods for client access:

- BasicAuth: User:Password

- BasicAuth: User:PassTicket

Communication between the Policy Services Provider and Customization Dialog requires authentication and authorization. These processes are implemented by using the z/OS PassTickets when a target TSO user makes a policy service request, for example, a JSON-report request on the Customization Dialog.

For environments that use User:Password authentication, no additional configuration is required.

However, when using User:PassTicket authentication, additional RACF setup is necessary to enable PassTicket generation and validation. For detailed setup instructions, see “Authentication using PassTickets and Authorization” on page 158.

To complete the necessary security setup steps, see “Authentication using PassTickets and Authorization” on page 158 in the "Security and Authorization" chapter.

(Optional) TLS HTTPS Connection Enablement

This task is needed only if you want to enable the Policy Services Provider TLS HTTPS connection. If not, skip this task.

To complete the necessary security setup steps, see “TLS HTTPS Connection Enablement Using a Self-Signed or CA Certificate” on page 160 in the Security and Authorization chapter.

Next step

After you set up the authentication and authorization settings and optionally enabled TLS HTTPS connection, you can start the Policy Services Provider. See “Step 30E: Start and Stop the Policy Services Provider” on page 137.

Step 30E: Start and Stop the Policy Services Provider

After you set up the configuration file and JCL procedure and optionally enabled HTTPS connection, you can start the Policy Services Provider.

The Policy Services Provider (PSP) now runs in two separate address spaces, which must both be active for PSP to operate correctly:

- SA PSP REST Server — started using procedure INGEPSP (Java-based)
- SA PSP ISPF Bridge — started using procedure INGEPSPI (C-based)

Both components work together the REST server handles API requests, and the ISPF bridge connects PSP to ISPF dialogs

Starting the Policy Services Provider

You must start both address spaces to activate PSP completely.

Step 1: Start the REST Server

Issue the following START console command on the system command line or syslog panel:

```
START INGEPSP,JOBNAME=<jobname1>
```

Expected messages (SA Customization REST Server)

Startup and Normal operation messages:

Message ID	Message Text
ING940I	IBM Z System Automation Policy Services Provider is initializing.

Message ID	Message Text
ING941I	IBM Z System Automation Policy Services Provider is initialized.

Termination and Error messages

Following are the error messages:

Message ID	Message Text
ING942I	IBM Z System Automation Policy Services Provider is terminating.
ING943I	IBM Z System Automation Policy Services Provider is terminated.
ING944I	IBM Z System Automation Policy Services Provider is breaking.
ING945E	IBM Z System Automation Policy Services Provider is broken: *&1*

Refer to the syslog and the Java log (<config-dir>/log/java.out) for further details.

Step 2: Start the ISPF Bridge

Issue the following **START command**:

```
START INGEPSPI,JOBNAME=<jobname2>
```

Expected Messages (ISPF Bridge)

Startup and Normal Operation Messages

Message ID	Message Text
ING930I	IBM Z System Automation Policy Services Provider ISPF Bridge is initializing.
ING931I	IBM Z System Automation Policy Services Provider ISPF Bridge is initialized.

Termination and Error Messages

Message ID	Message Text
ING932I	IBM Z System Automation Policy Services Provider ISPF bridge is terminating.
ING933I	IBM Z System Automation Policy Services Provider ISPF bridge is terminated.
ING934I	IBM Z System Automation Policy Services Provider ISPF bridge is broken.
ING935I	ISPF service failed with table table-name and function function-name.
ING936I	User userid is not authorized to use Policy Services Provider.

Message ID	Message Text
ING937I	User userid requested access access to resource profile profile-name in resource class class, but access is forbidden.

Check the syslog for related diagnostic messages to identify and fix the issue.

Stopping the Policy Services Provider

To stop the Policy Services Provider, both address spaces must be stopped.

Step 1: Stop the REST Server

Issue the **STOP** command for the PSP REST Server job:

```
STOP <jobname1>
```

Expected Messages (Java REST Server)

Message ID	Message Text
ING942I	IBM Z System Automation Policy Services Provider is terminating.
ING943I	IBM Z System Automation Policy Services Provider is terminated.

Step 2: Stop the ISPF Bridge

Issue the **STOP** command for the ISPF Bridge job:

```
STOP <jobname2>
```

Expected Messages (ISPF Bridge)

Message ID	Message Text
ING932I	IBM Z System Automation Policy Services Provider ISPF bridge is terminating.
ING933I	IBM Z System Automation Policy Services Provider ISPF bridge is terminated.

Next step

After you completed all configurations steps (set up the USS directory, JCL procedure, security and have the Policy Services Provider started), you can generate the policy database report in JSON format. See "How to create a PDB report in JSON format" in the *Define Automation Policy* manual.

Step 31: Enable the End-to-End Automation and Connect an SAplex to IBM Z Automation Web Console

This step points you to the configuration of end-to-end components (E2E agent and E2E adapter), which are installed by default through SMP/E into the System Automation zFS data set, and how to connect your SAplex to Service Management Unite.

SysOps	ProcOps
*	

End-to-end automation and connecting your SAplex to Z Automation Web Console are part of the SAplex configuration, which is achieved by the Configuration Assistant. If you plan to use the Configuration Assistant to configure the end-to-end components, refer to Chapter 9, “Base SA z/OS Configuration Using the Configuration Assistant,” on page 51.

If you plan to perform the configuration steps manually, complete the steps described in *IBM Z System Automation End-to-End Automation*.

You can import the best practice policy, *E2E, which is delivered with System Automation, into your policy database and customize its definitions there to fit your environment.

Step 32: Copy and Update Sample Exits

About this task

SysOps	ProcOps
*	*

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 Z System Automation Customizing and Programming*.

Step 33: Install Relational Data Services (RDS)

About this task

SysOps	ProcOps
*	

If you plan to use Relational Data Services (RDS) an extra VSAM cluster needs to be defined in order to make RDS tables persistent.

The sample job INGEMUVS is provided in ING.SINGSAMP to define the VSAM cluster.

Adapt your NetView startup procedure and add DD statement:

```
//INGEMUGL DD DSN=#hlq#.#domain#.EMUGLBL,DISP=SHR
```

You may also refer to sample startup procedure INGENVSA in ING.SINGSAMP.

Note: Due to the maximum records size of 32000 for a VSAM KSDS record, a RDS table row cannot be larger than 32000 bytes.

Step 34: Install CICS Automation in CICS

SysOps	ProcOps
*	

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.

Step 34A: SIT or Startup Overrides

About this task

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)
```

If CICS is started with option MSGLVL=0, some of the messages may not be passed to automation.

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 34B: Program List Table Definitions

About this task

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
```

The EVESTISP program definition in this example is only needed when using the CICS PPI communication.

Add the TYPE=ENTRY definitions shown in the following example to the shutdown program list table (PLT) for each CICS.

```
DFHPLT TYPE=INITIAL,SUFFIX=yy
DFHPLT TYPE=ENTRY,PROGRAM=EVESPLTT
DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
DFHPLT TYPE=FINAL
```

The EVESPLTT program definition in this example is only needed when using the CICS PPI communication.

Assemble the PLT tables.

Step 34C: Define Consoles

About this task

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 Z System Automation Customizing and Programming* or *IBM Z System Automation Defining Automation Policy*.
2. The CONSNAME parameter on the PROFILE statement in the task profile determines the EMCS console name. For more information, see *IBM Z NetView Administration Reference* and *IBM Z NetView Security Reference*.
3. By default the autotask name is used for the EMCS console name.

What to do next

A console has to be defined for each SA z/OS work operator. These are typically named AUTWRK*nn*. 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 34D: Transaction and Program Definitions

About this task

This step describes how to define the standard CICS Automation transactions and programs to CICS. The DFHCSDUP program is used to do this.

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 a group called EVEGRP1.

Before you run it, modify the job as directed in the JCL comments.

When using the CICS PPI, run the EVESJPPI sample job to define the necessary transactions and programs in a group called EVEGRP2.

Step 34E: DFHRPL and the CICS Automation Library

About this task

Update the DFHRPL concatenation to add the ING.SINGLOAD 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 34F: Add Libraries to NetView

About this task

Uncomment any libraries that you require in the INGENVSA member of the SINGSAMP data set. Refer to the sample for more details.

Step 34G: Installing CICSPlex SM REXX API

About this task

The CICSPlex System Manager REXX API is required for the interaction between SA z/OS and the CICSPlex System Manager. The REXX runtime interface to the API is supplied as a function package or host command environment. It should preferably be added to the function package table in the NetView module DSIRXPRM, as shown in "Step 6E: Add the REXX Function Packages to DSIRXPRM" on page 84.

For details about the installation of a function package, see *CICS Transaction Server for z/OS Installation Guide* and *IBM Z NetView Tuning Guide*.

Step 35: Install IMS Automation in IMS

SysOps	ProcOps
*	

Step 35A: Specify Required Control Region Parameters

About this task

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.

Note: Depending on your security requirements and authority assignments, CMDMCS can also be set to values of R, C, or B. For more information, refer to the *IMS System Definition Reference*.

Modify the IMS DBCTL control region JCL to specify the following parameter:

PREMSG=N
This is required for correct operation of IMS Product Automation.

Note: If PREMSG=Y is selected, all system messages and command responses are issued as multi-line messages. The first line is: `DFS000I MESSAGE(S) FROM ID=XXXX` where XXXX is the IMSID. The message starts on the second line. As a result, IMS message automation will not work as expected.

Step 35B: Install DFSAOE00 Exit

About this task

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.SINGLOAD 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 Z System Automation Product Automation Programmer's Reference and Operator's Guide* for further customization details.
- Use the exit that is supplied by SA z/OS on its own.
  - This involves concatenating the ING.SINGLOAD library after the IMS.SDFSRESL library in the STEPLIB concatenation, unless ING.SINGLOAD 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.SINGLOAD library after the IMS.SDFSRESL library in the STEPLIB concatenation, unless ING.SINGLOAD is in the linklist concatenation chain.
  - Call the EVIPVEX1 module from your exit program as detailed in *IBM Z System Automation Product Automation Programmer's Reference and Operator's Guide*.

This step is only required when you have made definitions in the MESSAGES/USER DATA policy against an IMS subsystem or class for IMS messages that need to be WTO'd.

Refer to "IMS Message Processing" in the *IBM Z System Automation Product Automation Programmer's Reference and Operator's Guide* for more details.

Step 35C: Add Libraries for NetView

About this task

Uncomment any libraries that you require in the INGENVSA member of the SINGSAMP data set. Refer to the sample for more details.

In order to issue IMS type 2 commands, access must available to the IMS modules, CSLSRG00 and CSLSDR00. These modules are shipped in the IMS product library named hlq.SDFSRESL. The entire product library can be allocated, or a private data set with just those modules and perhaps an explicit allocation or a LNKLST entry.

Step 36: Install ZWS Automation in ZWS

SysOps	ProcOps
*	

Step 36A: Add Libraries to ZWS

About this task

Add your SINGLOAD library and the NetView CNMLINK library containing CNMNETV to the ZWS steplib. Alternatively, you may add these libraries to LINKLST. You should have already APF-authorized these libraries.

Step 36B: Add Libraries to NetView

About this task

Allocate the EQQMLOG library according to your ZWS definitions. This data set contains any error messages that may occur when using the ZWS APIs on this NetView.

EQQMLIB should point to the appropriate message library for the level of ZWS 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 36C: Update ZWS Parameters and Exits

About this task

If you use the 'command request interface' that is based on automation workstations, you have to install the exit module EQQUXSAZ.

If you use the 'Conventional Request Interface' that uses general workstations named NV*xx*, the ZWS exit EQQUX007 has been configured or installed. A recycle of ZWS 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 ZWS Automation.

ZWS 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
UX011001
```

EQQUX007 and EQQUX011 are the exit driver programs. They call other modules in turn, as though ZWS 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 ZWS 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 ZWS.

You must specify the CALL07(YES) parameter in the ZWS z/OS initialization parameters.

You must specify the CALL11(NO) parameter in the ZWS z/OS initialization parameters if you want to monitor CP deletes. CP delete monitoring allows ZWS Product Automation to clear outstanding SDF alerts when an application or operation is deleted from the current plan. However, use of this exit will increase the CPU used by ZWS z/OS.

Other initialization parameters must be specified in the ZWS initialization member (EQQPARM) so that ZWS will issue some of its messages to the MVS console.

The DURATION, ERROROPER, LATEOPER, and OPCERROR messages are automated by ZWS 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 ZWS-supplied message members for certain messages.

The following messages are automated and may require changes to the ZWS-supplied message members in the SEQQMSG0 data set:

Table 21. SEQQMSG0 Data Set

Member	Message
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 ZWS can be found in *IBM Z Workload Scheduler Customization and Tuning*.

Note: If you use SDF to monitor the status of ZWS 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 SDF alerts for the same ZWS event as a result of the following:

- 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 SDF if UX007004 is not active. This is because ZWS does not issue a message when these operations exit error status.

Step 37: Configuring GDPS

SysOps	ProcOps
*	

This section describes the necessary customization and definitions when running GDPS on top of SA z/OS.

You can also import the best practice policy, *GDPS, which is delivered with SA z/OS, into your policy database and customize its definitions there to fit your environment.

Step 37A: Preparing NetView

Procedure

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, copy them to your user data sets and make your modifications there.

   The GDPS controlling system 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: INGXSGxx, where xx is the value of &SYSCLONE. This corresponds to HSAPRMKS as described in “Step 37B: Preparing the Automation Manager” on page 147.
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.

   For a list of valid gdps-option statements, refer to the INGSTGEN member from the sample library.

   Note: If the TOWER.SA includes GDPS, the VPCEINIT installation exit that is required by each supported GDPS product is automatically called during initialization of SA z/OS. Additionally, System Automation will automatically disable recovery for minor resources MVSESA.CF and MVSESA.XCF. If the TOWER includes ACTIVEACTIVE and the TOWER.SA includes GDPSSAT, the VPCSINIT installation exit that is required by the GDPS AA Satellite product is automatically called during initialization of SA z/OS.

Step 37B: Preparing the Automation Manager

About this task

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 37C: Defining the Automation Table Used by GDPS

GDPS provides an automation table (AT) named GEOMSG01. It contains user fragments GEOMSGU1 and GEOMSGU2, as well as product provided fragments that contain AT message traps. The GEOMSG01 AT is loaded automatically when the tower GDPS is set.

You can use the following user AT fragments to handle message traps that are *supplied by GDPS*:

- GEOMSGU1 for messages that should not flow into the GDPS provided AT fragment.
- GEOMSGU2 for messages that do not have an entry in the GDPS provided AT fragment.

For messages that should be processed by a user AT as well as the GDPS ATs, you should use a separate AT that is activated in parallel. You can achieve this by specifying multiple AT members in the SYSTEM INFO policy of the system (entry type SYS).

Note: GDPS clients should also review appropriate GDPS documentation for MPFLSTxx recommendations.

Step 38: Installing Tivoli Enterprise Portal Support

About this task

SysOps	ProcOps
*	

Step 38A: Enabling SOAP over HTTPS for a TEMS

This step is necessary if you want SA z/OS to direct SOAP queries to Tivoli Enterprise Monitoring Server (TEMS) using the HTTPS protocol. If you do not do this, you can only use the insecure HTTP protocol.

If you intend to communicate with multiple TEMS servers (for example, in a HA hub TEMS configuration not running on z/OS) from the same system you need to repeat for each one.

Please refer to the z/OS Communication Server documentation for details.

Be aware that the TCP/IP profile has to contain the statement `TCPCONFIG  TTLS` to result in the activation of the processed policy definitions.

AT-TLS Policy

Figure 10 on page 149 is a sample AT-TLS policy with the highest TCPIP trace. Please specify <tlsKeyring> and <ip_addr> accordingly. The <ip_addr> is the IP address of the machine hosting the TEMS server that you wish to direct the SOAP query to:

```
TTLSRule                            NV_TEMS_WIN
{
    LocalAddr                          ALL
    RemoteAddrRef                      addr_TEMS
    LocalPortRange                        0
    RemotePortRange                    3661
    Direction                          Outbound
    Priority                           255
    TTLSGroupActionRef                 XXGRP
    TTLSEnvironmentActionRef           XXENV
    TTLSConnectionActionRef            XXCON
}
TTLSGroupAction                     XXGRP
{
    TTLSEnabled                        On
}
TTLSEnvironmentAction               XXENV
{
    HandshakeRole                      Server
    EnvironmentUserInstance            0
    TTLSKeyringParmsRef                keyRing
    TTLSEnvironmentAdvancedParmsRef    XXADV
    Trace                              255
}
TTLSConnectionAction                XXCON
{
    HandshakeRole                      Client
    Trace                              255
}
TTLSEnvironmentAdvancedParms        XXADV
{
    ApplicationControlled              Off
  ClientAuthType                    PassThru
}
TTLSKeyringParms                    keyRing
{
    Keyring                            <tlsKeyring>
}
IpAddr                              addr_TEMS
{
    addr                               <ip_addr>
}
```

Figure 10. Sample AT-TLS policy

Certificate registration in keyring

The ITM Soap Server sends a self-signed certificate which has to be registered in the keyring. The certificate can be obtained easily if a web request is sent from a workstation browser.

Use the following URL for this purpose:

```
https://<ip_addr>:3661///cms/soap/kshhsoap.htm
```

You are asked to accept or deny the certificate. Store this certificate in X.509 PEM format (base64), upload this file to z/OS with ASCII to EBCDIC translation and add it to your keyring.

Chapter 11. Security and Authorization

You can secure the product. Only authorized personnel are able to access product-specific data sets, find out runtime information about automated resources, or change the status of such resources.

After the initial configuration, the product is set up so that you familiarize yourself with the functions for testing purposes and you make it secure for your production environment. However, before you begin, you are advised to change the default passwords of the operator IDs that come with the product. You locate the default operators that are defined in <nv_hlq_smpe>.DSIPARM member DSIOPFEX. Copy this member to <sa_hlq_user>.DSIPARM, edit it and change the PASSWORD parameter for each of them. For example, to change OPER1's password to XYZ123, specify:

```
OPER1        OPERATOR    PASSWORD=XYZ123
             PROFILEN    DSIPROFA
```

Use a System Authorization Facility (SAF) product, such as the z/OS Resource Access Control Facility (RACF) to secure your environment as follows:

- Operators are defined and authenticated by a SAF product
- Command authorization is done by a SAF product that is based on the issuer of a command
- Resource authorization is done by a SAF product that is based on the issuer of particular commands

SA z/OS facilitates the steps of securing your environment. The Configuration Assistant generates the INGESAF member that is based on the input in your `Configuration Options` file. The INGESAF member contains the following items:

- Profiles that protect commands and other resources
- Definitions of groups that represent roles
- Group membership that contain the individual operators in each role
- Necessary definitions for all the auto operators that are required by the product
- PERMIT statements that grant certain roles access to definitions for commands

You find the INGESAF member and all the other generated members in the CONFLIB data set. See Chapter 9, “Base SA z/OS Configuration Using the Configuration Assistant,” on page 51 for details about using the Configuration Assistant.

It is assumed that you intend to follow the IBM recommendations to secure your automation environment, and to use the samples in the INGESAF member. See *IBM Z NetView Security Reference* for a complete description for details about the recommended settings and other security options that you can use.

Notes:

1. For evaluation and browsing purposes a member INGESAF in a readable format is also provided in the SINGSAMP sample library. Refer to the description section of this member and discover the provided security definitions within this member. For establishing the SAF-based security environment it is required to use the Configuration Assistant.
2. Make sure you have APAR OA41282 installed. With this APAR, the z/OS RACF provides the new general SYSAUTO resource class as a system-provided resource class.

   When using a SAF product other than RACF, manually define the SYSAUTO class.

Authorization of the Started Procedures

The started procedures for the Automation Manager, the Automation Agent, the Subsystem Interface, and the IPL Data Gatherer need authority to access SAF-protected resources.

Use the STARTED class. None of the started procedures requires the PRIVILEGED or TRUSTED attribute. You must check with your security administrator for details.

The names of the started procedures are listed in Table 22 on page 152:

Table 22. Started Procedure Names for Functions

Function	Default Procedure Name	Real Procedure Name
Automation Manager	INGEAMSA	Value of `sa_am_start_proc`, otherwise use what is specified for `sa_am_start_job.1`.
Spare Automation Manager	INGEAMSA	Value of `sa_am_start_proc` , otherwise use what is specified for `sa_am_start_job.2`.
Automation Agent	INGENVSA	Value of `sa_saagent_start_proc`, otherwise use what is specified for `sa_saagent_start_job.`
Subsystem Interface	CNMSJ010	Value of `sa_nvssi_start_proc`, otherwise use what is specified for `sa_nvssi_start_job`.
IPL Data Gatherer	HSAPIPLC	Value of `sa_ipldata_start_proc`, otherwise use what is specified for `sa_ipldata_start_job`.
E/AS for SA z/OS event notification	IHSAEVNT	Value of nv_eas_start_proc, otherwise use what is specified for `nv_eas_eif_start_job`.
E/AS for E2E adapter infrastructure	IHSAEVNT	Value of nv_eas_start_proc, otherwise use what is specified for `nv_eas_e2e_start_job`.
E2E adapter	INGXADPT	Value of sa_e2eadpt_start_proc, otherwise use what is specified for `sa_e2eadpt_start_job`.
E2E agent	INGXEAGT	Value of sa_e2eagnt_start_proc, otherwise use what is specified for `sa_e2eagnt_start_job`.

Table 23 on page 152 lists the SAF-protected resources, that each started procedure needs access to:

Table 23. SAF-protected Resources for Functions

Function	SAF-Resources	Access
Automation Manager	`<sa_automation_policy>`.SOCNTL	READ
	`<sa_hlq_smpe>`.**	READ
	`<sa_hlq_user>`.**	UPDATE

Table 23. SAF-protected Resources for Functions (continued)

Function	SAF-Resources	Access
Automation Agent	<sa_automation_policy>.SOCNTL	READ
	<sa_hlq_smpe>.**	READ
	<nv_hlq_smpe>.**	READ
	<sa_hlq_user>.**	READ
	<sa_hlq_user>.*.DSILIST	UPDATE
	<sa_hlq_user>.*.STATS	UPDATE
	<sa_hlq_user>.*.DSILOG%	UPDATE
	<sa_hlq_user>.*.DSIDVRT	UPDATE
IPL Data Gatherer	<sa_hlq_user>.INGXSG*.IPLDATA	CONTROL
	SYS1.PARMLIB	READ
Subsystem Interface	<sa_hlq_smpe>.**	READ
E/AS for SA z/OS Event notification	<hlq_user>.SCNMUXCL	READ
	<nv_hlq_smpe>.**	READ
E/AS for E2E adapter infrastructure	<hlq_user>.SCNMUXCL	READ
	<nv_hlq_smpe>.**	READ
E2E adapter	<sa_usspath_smpe>	READ/EXECUTE
	<sa_usspath_user>/<sa_usspath_user_e2e>/config	READ
	<sa_usspath_user>/<sa_usspath_user_e2e>/ssl	READ
	<sa_usspath_user>/<sa_usspath_user_e2e>/data	READ/WRITE
E2E agent	<sa_usspath_smpe>	READ/EXECUTE
	<sa_usspath_user>/<sa_usspath_user_e2e>/config	READ
	<sa_usspath_user>/<sa_usspath_user_e2e>/ssl	READ
	<sa_usspath_user>/<sa_usspath_user_e2e>/data	READ/WRITE

To enable the Automation Manager to properly shut down OMVS, super user permission for UNIX System Services must be granted. The Automation Manager's user must have an OMVS segment and access to the BPX.SUPERUSER resource.

Additionally, add library <sys_hlq_sceerun>.SCEERUN, <sys_hlq_sceerun>.SCEERUN2, <sys_hlq_sceerun>.CSSLIB, and <sys_hlq_sceerun>.SCLBDLL as Program Controlled and authorize the Automation Manager's user accordingly. Check with your security administrator for details.

Roles

To facilitate the definition of command authorizations for human and auto operators, it is recommended to use groups. Each group corresponds to a certain usage profile or role.

The product comes with five predefined roles that are described in Table 24 on page 154:

Table 24. Security Roles

Role	Default Group	Description
User	INGUSER	In this role, an operator can merely display a few things but cannot change or otherwise influence how the automation works.
Operator	INGOPER	In this role, an operator can use panels to do what is necessary to keep the system in running order on a day to day basis.
Administrator	INGADMIN	In this role, an operator has the rights to perform special commands. Such as loading of a new automation configuration or otherwise act beyond the scope of the daily work of a normal operator.
Auto Operator	INGAUTO	In this role, an operator has the rights to perform all commands and services that can be started from the product, user scripts, and the automation table that are required to bring resources into their Desired status or to recover from failures. The permissions for this role are required for the product to work correctly as specified in the INGESAF member.
	INGWRK	INGWRK is not a role. It is a functional group hosting additional permissions for a subset of auto operators.
Superuser	INGSUPER	In this role, an operator has no restrictions.

Refer to the INGESAF member for a complete reference of commands and services and the associated roles as provided by the product.

Note: The mapping of roles and commands in the INGESAF member is only a guideline. Following the recommendations in this member, however, reduces the time to secure the environment.

If you have groups in your environment that you would like to reuse, ensure that the groups are defined with similar characteristics as described in the INGESAF member. In particular, if you intend to automate UNIX System Services processes, for each group, an OMVS segment is required that contains the group ID for this group.

The following variables in the `Configuration Options` file are used to specify the SAF-group name for each of the roles that are listed here. You do not have to change the default names unless your organization follows a naming convention:

Table 25. Option File variables for SAF-group name

Options File variable	Default value
`racf_group_user`	INGUSER
`racf_group_oper`	INGOPER
`racf_group_admin`	INGADMIN
`racf_group_auto`	INGAUTO
`racf_group_autowrk`	INGWRK
`racf_group_super`	INGSUPER

The following variables in the `Configuration Options` file are used to specify the USS group IDs of the groups that are listed here. You do not have to change the default unless there are conflicting assignments for other groups. Check with your security administrator.

Table 26. Option File variables for UNIX System Services Group IDs

Options File variable	Default value
racf_omvs_gid_user	80002
racf_omvs_gid_oper	80003
racf_omvs_gid_admin	80004
racf_omvs_gid_auto	80001
racf_omvs_gid_autowrk	80006
racf_omvs_gid_super	80005

To associate human operators to these groups, the `Configuration Options` file provides the `<racf_group_xxxxx>` variables, where you can specify which operators are members of a group. If you use these variables, the Configuration Assistant automatically generates the appropriate RACF statement in the INGESAF member. Otherwise, your security administrator has to connect these operators to the groups manually.

For example, to associate operator BOB with the INGUSER group and operators GABI and TIM with the INGOPER group, the following variable must be specified in the `Configuration Options` file:

```
racf_group_user=INGUSER:BOB
racf_group_oper=INGOPER:GABI,TIM
```

Note: It is not necessary to specify auto operators as the generated definitions in the INGESAF member define each auto operator with a default group as specified in variable `<racf_group_auto>`.

Any data set access permissions that are required for all operators that are connected to these groups are provided automatically in the INGESAF member.

Operators

All operators, human and automated operators, are defined and authenticated by an SAF product.

For example, to define a human operator who is called BOB with RACF, the following definition is needed:

- A NetView segment must be created.
  `ALU BOB NETVIEW(IC(LOGPROF1) MSGRECVR(NO) CTL(GLOBAL))`

- Data set permissions must be granted.

  Note: If you use the Configuration Assistant and follow the IBM recommendations, the granting of permissions is accomplished implicitly through group membership and group permissions as defined in the generated INGESAF member. See also the previous subsection.

- (Optional) An OMVS segment must be created if you want to automate UNIX System Services processes
  `ALU BOB OMVS(UID(uid) HOME('/u/bob') PROGRAM ('/bin/sh'))`

  Where `uid` is a 1 - 10 digit integer value. It is the responsibility of your Security Administrator to define the human operators, appropriately.

A human operator might have other related SAF attributes, such as a default group it belongs to, a default data set profile, a TSO segment, and other information that is out of the scope of this document.

Note: You do not have to make the definitions for the auto operators yourself. The INGESAF member contains all the RACF commands that are necessary to add a user and set the necessary characteristics. Included is the definition of an OMVS segment and read access to BPX.SUPERUSER for those auto operators that can automate USS processes.

Finally, the SECOPTS.OPERSEC stylesheet option has to be set like follows:

```
SECOPTS.OPERSEC = SAFDEF
```

See also section “Stylesheet Options” on page 167 for more information.

Commands

All commands and services that can be used by human operators and auto operators are protected by an SAF product.

If you use the Configuration Assistant and follow the IBM recommendations, nothing specific has to be done by you. The INGESAF member contains all the RACF commands necessary to define profiles and permissions on a group, that is, role basis.

For your reference, the profiles that are specified for SAF class NETCMDS are constructed with the following pattern:

```
netid.domain.command
```

The generated statements in the INGESAF member use wildcards. However, you can use wildcards for the variables here, only when the NETCMDS class has generics enabled. To enable generics, through RACF you can use the following command:

```
SETR GENERIC(NETCMDS)
```

The sample profile definitions in the INGESAF member do not allow for the use of all commands (product-provided and user scripts) in general. But because there are many commands that can be considered "safe" in the automation environment, it also grants all defined user roles access to all commands that are not explicitly listed in the INGESAF member. Thus, you avoid having an explicit profile that is defined for each of those commands. With RACF, the definitions in the INGESAF member look like this example:

```
RDEFINE NETCMDS *.*.* UACC(NONE)
RDEFINE NETCMDS *.*.* ID(INGUSER,INGOPER,INGADMIN,INGAUTO,INGSUPER) UACC(READ)
```

The INGESAF member lists profiles that allow for the use of commands that are based on roles. For more information about using the NETCMDS class, see *IBM Z NetView Security Reference*.

The INGESAF member is a sample member that implements the mapping of commands to roles as recommended by the product. Your security administrator can take the generated commands as they are. Or adjust as needed, for example to add or remove certain groups for a particular command.

Finally, the SECOPTS.CMDAUTH stylesheet option has to be set like follows:

```
SECOPTS.CMDAUTH = SAF,PASS
```

See also section "Stylesheet Options" on page 167 for instructions.

Use of Commands Cross System

All operator commands that are provided by the product supply a TARGET parameter that you can use to run the command on a remote system.

SA z/OS is limited to just a small number of systems in the sysplex by default. The INGESAF member contains a definition that explicitly allows the communication between those systems.

However, if you want to limit this capability or completely prevent that commands can be issued on one system but run on another system then more profiles and permission statements are required. The profiles are defined in the SYSAUTO general resource class and constructed according to the following pattern:

AGT.*sysplexname*.*saxcfgroup*.TARGET.FROMDOM.*fromdom*.TODOM.*todom*

The variables have the following meanings:

sysplexname
This variable denotes the name of the physical sysplex.

saxcfgroup
This variable denotes the XCF group name for this particular system. The name always starts with the prefix INGXSG, followed by the value that is specified in the <sa_xcf_grpid_suffix> variable in the Configuration Options file.

fromdom
This variable denotes the NetView domain on which commands with the TARGET parameter can be issued.

todom
This variable denotes the NetView domain on which commands with the TARGET parameter can be run.

You can use wildcards in the profile, when the class SYSAUTO has generics enabled. To enable generics, with RACF you can use the following command:

```
SETR GENERIC(SYSAUTO)
```

The following profile can be defined to prevent execution of commands on the IPUFA domain. Using RACF, for example, the command looks like the example here:

```
RDEFINE SYSAUTO AGT.*.*.TARGET.FROMDOM.*.TODOM.IPUFA UACC(NONE)
```

In order to allow BOB to run commands on domain IPUFA, the following permission statement can be used:

```
PERMIT AGT.*.*.TARGET.FROMDOM.*.TODOM.IPUFA CLASS(SYSAUTO) ID(BOB) ACC(READ)
```

Notes:

1. The read access to such a profile enables execution of a command on a remote system only when the issuer is authorized to start the command on the local system.
2. For the security checks to prevent unauthorized use of commands across systems, it is important that the SYSAUTO class is activated, a profile exists, and that the SAF-product is active. If a check fails indicating that any of these conditions is not met, access is granted, regardless.

Securing the Policy Services Provider

This topic describes how to properly set up the security environment for the Policy Services Provider.

Authentication using PassTickets and Authorization

It's required to enable PassTicket support for the Policy Services Provider and to authorize users to obtain PassTickets before they can request services from Policy Services Provider.

About this task

The RACF PassTicket is a one-time-only password that is generated by a requesting product or function (System Automation in this case). It is an alternative to the RACF password and password phrase that removes the need to send RACF passwords and password phrases across the network.

The Policy Services Provider is an ISPF background application that provides policy services requested from the Customization Dialog users, for example, a request to generate a PDB report in JSON format. The required authentication and authorization between Policy Services Provider and Customization Dialog is implemented by PassTickets as shown in this following flow:

1. A user logged on to the TSO attempts to call a Policy Services Provider service from the Customization Dialog.
2. System Automation generates a PassTicket for the application name INGPSP and passes it with the request to the Policy Services Provider.
3. The Policy Services Provider receives the PassTicket from the request and eventually authenticates the user and verifies the user's authorization. If the user passes the authentication and authorization, Policy Services Provider fulfills the service request.

IBM recommends using enhanced PassTickets for authentication. For more information about enabling enhanced PassTickets, see Using PassTickets in the z/OS manual *z/OS Security Server RACF Security Administrator's Guide*.

Authentication using PassTickets

To use the z/OS PassTickets function, complete the following setup activities.

1. System Automation delivers the authorized TSO INGPGPTK program. The Policy Services Provider requires that the INGPGPTK program must be defined as an authorized program in TSO. This can be achieved by adding the INGPGPTK program name to the PARMLIB member IKJTSO*xx* in SYS1.PARMLIB under **AUTHTSF**.

   Use the TSO/E command **PARMLIB UPDATE(xx)**, or MVS command **SET IKJTSO=xx** to activate the new settings. Be sure that INGPGPTK is concatenated in the LINKLIST.
2. Activate the security class PTKTDATA. Optionally, if you want to use generic profiles, specify the **GENERIC** option with the **SETROPTS** command, too.

   ```
   SETROPTS CLASSACT(PTKTDATA) RACLIST(PTKTDATA) GENERIC(PTKTDATA)
   ```
3. Create an application profile for the application INGPSP in class PTKTDATA. Specify the PassTicket key in the profile. You can use the following three possible approaches.
   - Recommended: Enhanced PassTicket support using an HMAC secret key that is encrypted and stored in ICSF.

     ```
     RDEFINE PTKTDATA INGPSP SSIGNON(EPTKEYLABEL(MY.HMAC.KEY))
     ```

- Legacy: The PassTicket key is kept in the RACF database by specifying the secret key as value of the **KEYMASKED** keyword.

```
RDEFINE PTKTDATA INGPSP SSIGNON(KEYMASKED(0123456789ABCDEF))
```

- Alternatively, the PassTicket key can be encrypted and additional authorization steps are required to proceed with this alternative that aren't explained here. In this case, the KEYENCRYPTED value is specified as PassTicket key. Here is an example:

```
RDEFINE PTKTDATA INGPSP SSIGNON(KEYENCRYPTED)
```

4. For the settings to become active, refresh the PTKTDATA class in the RACF settings by using the **SETROPTS** command.

```
SETROPTS RACLIST(PTKTDATA) REFRESH
```

5. Every target user that needs to work with the Policy Services Provider needs to generate a PassTicket. The service is authorized and therefore is protected by using a resource profile in class SYSAUTO. Create a new profile in SYSAUTO with universal access NONE.

```
RDEFINE SYSAUTO TSO.sysplexname.systemname.PASSTICKET.INGPSP UACC(NONE)
```

   Next, grant every Policy Services Provider target user access to this profile to authorize them to generate a PassTicket.

```
PERMIT TSO.sysplexname.systemname.PASSTICKET.INGPSP CLASS(SYSAUTO) ACCESS(READ)
ID(target_user)
```

   Note: The profile name is leaned towards the existing pattern of other similar TSO-related profiles in class SYSAUTO. It allows you to create different profiles by sysplex and by system name. An asterisk * can be used as a wildcard to create a generic profile.

   Additionally, to the RACF definitions for the PSP the application needs the following RACF permission to check if a PassTicket is valid: `IRRPTAUTH.application.target-userid`

   Example:

```
PERMIT IRRPTAUTH.INGPSP.<user> CLASS(PTKTDATA) ACCESS(READ)
```

Authorization

To access the SA Customization API, the user ID must have READ access to the RACF resource class INGPSP.REPORT.

1. Before the Policy Services Provider fulfills a JSON-report request, the caller's authorization is verified. The report service is protected using a resource profile in class SYSAUTO. Create a new profile with universal access NONE.

```
RDEFINE SYSAUTO INGPSP.REPORT UACC(NONE)
```

   Next, grant every Policy Services Provider target user who should be able to request report services access to this profile.

```
PERMIT INGPSP.REPORT CLASS(SYSAUTO) ACCESS(READ) ID(target_user)
```

2. Grant the Policy Services Provider access to the policy database partitioned data set, so that it can access the database for reporting.

```
PERMIT profile-name CLASS(DATASET) ACCESS(READ) ID(ingpsp_user)
```

3. Finally, activate the definitions above with the **SETROPTS** command.

```
SETROPTS RACLIST(SYSAUTO) REFRESH
```

After all updates are made in RACF, PassTickets can be generated and evaluated. Every target user that needs to work with the Policy Services Provider can use the PassTicket to authenticate their user IDs.

If authentication failed, ING936I message is issued in the syslog.

```
ING936I User user is not authorized to use Policy Services Provider. Reason: ssrrnnnn.
```

If authorization fails, ING937 message is issued in the syslog.

```
ING937I User user requested access access to resource profile profile-name in resource
class class, but access is forbidden. Reason: ssrrnnnn.
```

For more information about ING936I and ING937I messages, see the *IBM Z System Automation Messages and Codes* manual.

TLS HTTPS Connection Enablement Using a Self-Signed or CA Certificate

This topic describes how to enable Policy Services Provider TLS HTTPS connection with either a self-signed or certificate authority (CA) certificate. If you don't plan to enable TLS HTTPS connection, skip this topic.

- "Use a self-signed certificate to enable TLS HTTPS connection" on page 160
- "Use a CA certificate to enable TLS HTTPS connection" on page 161

Use a self-signed certificate to enable TLS HTTPS connection

1. Create a self-signed certificate (.jks) by using the JDK keytool command-line tool. In this example, the created keystore is named `pspcer.jks`.

   ```
   keytool -genkey -alias pspcer -keypass 123456 -keyalg RSA -keysize 2048
   -validity 365 -keystore pspcer.jks -storepass 123456 -ext san=dns:example.mydomain.com
   ```

   Note: A self-signed certificate needs to contain proper DNS information (hostname). When you create the certificate, specify the DSN information with the **`-ext san=dns:`*`hostnname`*** option. If you omit it, you may encounter certificate validation errors such as '`checkServerCert: Certificate not valid for IP`' or '`checkServerCert: Certificate not valid for DNS name`'.
2. Upload the keystore certificate file in binary mode into the Policy Services Provider USS directory that is customized in "Step 30A: Set up the USS Directory Structure" on page 135.
3. Retrieve a PEM format of the certificate from keystore by using the keytool command-line tool. For example:

   ```
   keytool -export -rfc -alias pspcer -file pspcer.pem -keystore pspcer.jks -storepass 123456
   ```

   Check the PEM-format certificate content. It should display as the following example. If it contains messy characters, it means that the certificate is not correct and cannot be added into the RACF database in the later step.

   ```
   -----BEGIN CERTIFICATE-----
   MIIDOzCCAiOgAwIBAgIEIp5SUDANBgkqhkiG9w0BAQsFADBOMQswCQYDVQQGEwIw
   MDELMAkGA1UECBMCcHAxCzAJBgNVBAcTAmNpMQswCQYDVQQKEwJrazELMAkGA1UE
   CxMCa2sxCzAJBgNVBAMTAlVOMB4XDTIxMTIyOTAyMjYyN1oXDTIyMTIyOTAyMjYy
   N1owTjELMAkGA1UEBhMCMDAxCzAJBgNVBAgTAnBwMQswCQYDVQQHEwJjaTELMAkG
   A1UEChMCa2sxCzAJBgNVBAsTAmtrMQswCQYDVQQDEwJVTjCCASIwDQYJKoZIhvcN
   AQEBBQADggEPADCCAQoCggEBAJMJu3A5mShYq/8ID3Z0+uFx0jDlniqoBZ8zSCVu
   2CojynvDsCtcZWX6bYwGC9SoHP6L45iZR0GUUG9Ju6uD86fGNFtbHi8HXlWb+eGR
   rsXatEqCuiX6FtJKbjQbTWyb1H/+rQBeWszzEtzeg/3IENMpsGHU/PE5h5HC1s+a
   I0qDUIpRxI2UP0wHMu6UPBiRTtKJ4A+3nxWpVGQOlnivfIGg0NAQhyAIZ0E3gq0N
   QyNIWiIjmHDiRtXae8XWO1lbzVLQWsbx4jjJNwK6i6mkqqcffUO/5J8AE+Whvcye
   9T6c1/OmhwICalBobh/gaaCgPT2f+UxN8UhSSuF27DKL1/UCAwEAAaMhMB8wHQYD
   VR0OBBYEFJhDzcE3UvbPYdmbyUHUjH8BrNxMMA0GCSqGSIb3DQEBCwUAA4IBAQBJ
   U6Mwlnae9EUnRtGgFfd+2teFo4GnS9AN5rGpq+rPRO28lvGny8ushhlts6PVg5Nq
   +zo47uLnVDvKBq+LXyBdytsuBTsQnB3xqj0tY/KZX2kWy1DGcuGhJ/FcXjVUrIDH
   s0tyLWnGh6L4KucQw5uXqJdk5oSzSWUavb39Ou24qJn/92lwD45Hri8VOgIWb70A
   Hbwvuk5wqme6mEJpXOM2nxg0+Yv+p9B5AhBs10whZ56ryLrWjN7m2Cbc5U9GNRYW
   fY1lCvnxYJJupLPzQc/0hDonrNmB5gCKWLDMLKKvCGewzbbqLTHQ95EoTFNdfEuu
   FU+FxrJwMgC5dHtk3dIa
   -----END CERTIFICATE-----
   ```

4. Put the PEM-format certificate content into a sequential data set. When allocating the data set, you can set the **Record format** to VB and **Record length** to 255 (You can also try other setting here, but this setting is tested by the Z System Automation team).
   - You can copy and paste the certificate contents into the data set.
   - Or you can upload the certificate into the Policy Services Provider USS directory and then use the TSO **OGET** command to put the certificate content into the data set. As shown in this example command, specify your USS directory and the target data set (for example, `ING.PSP.PEM`).

   ```
   OGET '<config-dir>/config/security/pspcer.pem' 'ING.PSP.PEM'
   ```

5. Add the data set that contains the certificate content into the RACF database. The following example is for a personal self-signed certificate that can be used only by a single user.
   a. Add the data set ('`ING.PSP.PEM`' in this example) to RACF database for a TSO user (for example, PSPUSER) with a label (for example, '`PSP Server`'). For more information, see RACDCERT (Manage RACF digital certificates) in z/OS documentation.

   ```
   RACDCERT ID(PSPUSER) ADD('ING.PSP.PEM') TRUST WITHLABEL('PSP Server')
   ```

   b. Create a new key ring name (for example, DIALOG) for the TSO user.

   ```
    RACDCERT ADDRING(DIALOG) ID(PSPUSER)
   ```

   c. Connect the certificate to the key ring.

   ```
   RACDCERT ID(PSPUSER) CONNECT(LABEL('PSP Server') RING(DIALOG))
   ```

   d. Refresh the digital certificates definitions.

   ```
   SETROPTS RACLIST(DIGTCERT) REFRESH
   ```

   e. Verify the certificate and key ring definitions with the **RACDCERT** commands.

   ```
   RACDCERT ID(PSPUSER) LIST(LABEL('PSP Server'))
   RACDCERT ID(PSPUSER) LISTRING(DIALOG)
   ```

Use a CA certificate to enable TLS HTTPS connection

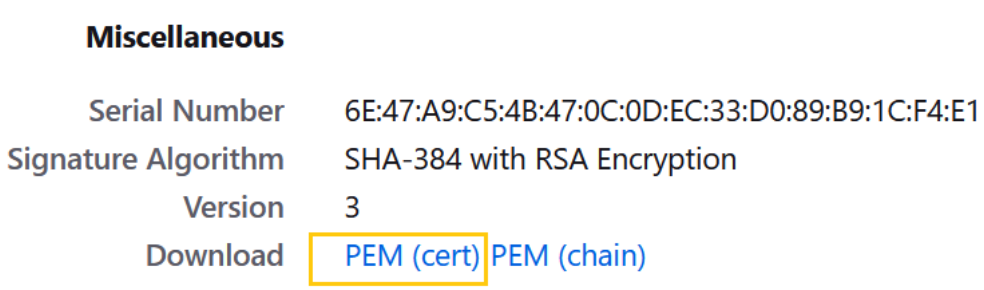
1. Import CA-signed certificates that are obtained from a certificate authority vendor into your existing keystore. In this example, three certificate files were received in total: `cert.der`, `caintermediatecert.der`, and `carootcert.der`. You need to concatenate them into one file (for example, `chain.der`) in the leaf-to-root order before importing them to the keystore. For example, use the following command.

   ```
   cat cert.der caintermediatecert.der carootcert.der > chain.der
   ```

   Import the `chain.der` file into your existing keystore, for example:

   ```
   keytool -keystore pspcert.jks -import -trustcacerts -alias pspcert -file chain.der
   -storepass storepass_value
   ```

2. Upload the keystore file in binary mode into the Policy Services Provider USS directory.
3. Access the Policy Services Provider host address with a browser and view the CA-certificates in the browser (for example, here is a guidance of how to view the certificate in Firefox; the other browsers might differ). Download the PEM (cert) file for each certificate one-by-one.



Note: If you download or receive only one file that contains multiple chained certificates, you must split it into separate files. One file matches with one certificate.

4. Create multiple sequential data sets (for example, `ING.PSP.SERVER.PEM`, `ING.PSP.ROOT.PEM`, and `ING.PSP.INTERMED.PEM`) to keep each PEM-format certificate one by one. You can copy and paste the certificate contents to each data set, or upload them to the USS directory and then use the TSO **OGET** command to put the contents to each data set accordingly.
5. Add all certificates data sets to the RACF database. For more information, see RACDCERT (Manage RACF digital certificates) in z/OS documentation.

```
RACDCERT SITE ADD('ING.PSP.SERVER.PEM') TRUST WITHLABEL('PSP SERVER CA')
RACDCERT CERTAUTH ADD('ING.PSP.ROOT.PEM') TRUST WITHLABEL('PSP ROOT CA')
RACDCERT CERTAUTH ADD('ING.PSP.INTERMED.PEM') TRUST WITHLABEL('PSP INTERMEDIATE CA')
```

6. Create a new key ring name (for example, PSPSRV) for the user (for example, PSPUSER). In this example procedure, multiple users share the keyring name within one environment (as further defined in step 10.)

```
RACDCERT ADDRING(PSPSRV) ID(PSPUSER)
```

7. Connect all certificates to key ring. Specify **CERTAUTH** for CA certificates.

```
RACDCERT CONNECT(CERTAUTH LABEL('PSP ROOT CA')
  RING(PSPSRV) USAGE(CERTAUTH)) ID(PSPUSER)
RACDCERT CONNECT(CERTAUTH LABEL('PSP INTERMEDIATE CA')
  RING(PSPSRV) USAGE(CERTAUTH)) ID(PSPUSER)
RACDCERT CONNECT(SITE LABEL('PSP SERVER CA')
  RING(PSPSRV) USAGE(PERSONAL) DEFAULT) ID(PSPUSER)
```

8. Refresh digital certificates definition.

```
SETROPTS RACLIST(DIGTCERT) REFRESH
```

9. Verify certificates and key ring definition.

```
RACDCERT SITE LISTCHAIN('PSP SERVER CA')
RACDCERT LISTRING(PSPSRV)
```

10. Define a profile in class RDATALIB and give multiple users (for example, PSPUSER and SAUSER) READ access to key ring.

```
RDEFINE RDATALIB PSPUSER.CERTRING.LST UACC(NONE)
PERMIT PSPUSER.CERTRING.LST CLASS(RDATALIB) ID(PSPUSER) ACCESS(READ)
PERMIT PSPUSER.CERTRING.LST CLASS(RDATALIB) ID(SAUSER) ACCESS(READ)
RDEFINE RDATALIB PSPUSER.CERTRING.UPD UACC(NONE)
PERMIT PSPUSER.CERTRING.UPD ID(PSPUSER) ACCESS(READ) CLASS(RDATALIB)
PERMIT PSPUSER.CERTRING.UPD ID(SAUSER) ACCESS(READ) CLASS(RDATALIB)
```

Note: In this example procedure, the key ring for user PSPUSER is PSPSRV. For other users (SAUSER in this example), the key ring is PSPUSER/PSPSRV.

11. Activate and refresh class RDATALIB definition.

```
SETROPTS CLASSACT(RDATALIB) RACLIST(RDATALIB)
SETROPTS RACLIST(RDATALIB) REFRESH
```

Use of Commands from TSO or Batch

You must define profiles if you want to use:

- the batch command interface
- AT overwrite syntax checking of the customization dialog
- Relational Data Services from TSO
- other SA z/OS provided REXX functions running in TSO.

Permissions must be granted also to those profiles for all users that might require these capabilities.

Note: By default, a command cannot be issued from outside the NetView 3270 console or the system console. The security precaution is enforced when you use the Configuration Assistant and deploy the definitions in the generated INGESAF member.

Authorization is granted to all users, if the following checks are passed successfully:

- Front-end check that basically allows permissions or rejects a user regardless of the particular command that is used
- Back-end check that performs an authorization check inside the NetView program on behalf of the TSO or batch user that starts the command

Front-end Checking

The profiles for Front-end checking are defined in the SYSAUTO general resource class and are constructed according to the following pattern:

```
TSO.sysplexname.systemname.CMDRCVR.SEND
```

The variables have the following meanings:

sysplexname
This variable denotes the name of the physical sysplex.

systemname
This variable denotes the name of the system.

You can use wildcards, when the class SYSAUTO has generics enabled. To enable generics, with RACF you can use the following command:

```
SETR GENERIC(SYSAUTO)
```

For example, the following profile can be defined to prevent the issuing of any command from TSO or Batch on the SYS1 system. The RACF syntax would be as follows:

```
RDEFINE SYSAUTO TSO.*.SYS1.CMDRCVR.SEND UACC(NONE)
```

To allow BOB to issue commands on the SYS1 system, the following permission statement can be used:

```
PERMIT TSO.*.SYS1.CMDRCVR.SEND CLASS(SYSAUTO) ID(BOB) ACC(READ)
```

Note: Read access to such a profile enables issuing of a command on that system only, if the user also passes the back-end check and the NetView command check.

Using the TSO function INGRCRPC, you may want to distinguish command execution in USERTASK or AUTOTASK. The command execution in the user task BOB requires the following permissions:

```
RDEFINE SYSAUTO TSO.*.SYS1.CMDRCVR.SEND.USERTASK UACC(NONE)
PERMIT TSO.*.SYS1.CMDRCVR.SEND.USERTASK CLASS(SYSAUTO) ID(BOB) ACC(READ)
```

For a detailed discussion about command execution in USERTASK and AUTOTASK, see the "Security Considerations" section of the "Function INGRCRPC" topic in *Customizing and Programming*.

Back-end Checking

The profiles for Back-end checking are defined in the NETCMDS class and constructed as described in “Commands” on page 156. If you use the Configuration Assistant and follow the IBM recommendations, the profiles are already defined for you. The INGESAF member contains all the RACF commands that you need to define profiles and permissions on a group, that is, role basis.

However, unless the TSO user or the user that is associated with the Batch job is already connected to any of the groups that represent different user roles (see “Roles” on page 154), more definitions are required.

Batch Command Interface AOFRYCMD/EVJRYCMD with SERVER=*

- The user must be permitted to use the command EVJRVCMD (Batch only)

  EVJRVCMD is the Batch receiver command name and INGRYRU0 is the true name of the INGREQ command, rather than just the command synonym of INGREQ.
- The user must be permitted to use each command that wants to issue (TSO and Batch)

For example, to allow the RUNAUTO job that is associated with user BOB to issue the INGREQ command, with RACF, the following permission statements as shown are required:

```
PE *.*.EVJRVCMD CLASS(NETCMDS) ID(BOB) ACC(READ)
PE *.*.INGRYRU0 CLASS(NETCMDS) ID(BOB) ACC(READ)
```

If BOB executes a user written command, then the command itself and all imbedded NetView commands require in addition read access by the auto task (for example AUTCMD*nn*), which is used to run the command.

TSO Function INGRCRPC

The user must be permitted to use each command that he wants to issue from TSO via function INGRCRPC.

For example, to allow TSO user BOB to execute the MYCMD command, with RACF, the following permission statements are required:

```
PE *.*.MYCMD CLASS(NETCMDS) ID(BOB) ACC(READ)
PE *.*.MYCMD CLASS(NETCMDS) ID(AUTCMD01) ACC(READ)
```

As the second statement shows, the autotask (AUTCMD01 in this example) where the command is executed also needs the permission. If the command is executed under the security context of TSO user BOB, then no autotask is involved and the second permission statement is not needed. For more details, see the "Security Considerations" section of the "Function INGRCRPC" topic in *Customizing and Programming*.

Relational Data Services INGRCRDX

The user must be permitted to use the command INGRCRDS. For example,

```
PE *.*.INGRCRDS CLASS(NETCMDS) ID(BOB) ACC(READ)
```

AT Overwrite Syntax Checking for the Customization Dialog

The user must be permitted to use NetView command PIPE. For example,

```
PE *.*.PIPE CLASS(NETCMDS) ID(BOB) ACC(READ)
```

Notes:

1. To help you finding the true name for a command, search the INGESAF member for the synonym that you are looking for.

2. Read access to such a profile ensures that the user really is authorized to issue the command even though that user might not even be known to the NetView program and the command is instead issued by an auto operator.
3. For the security checks to prevent the unauthorized use of commands from TSO or batch, it is important that a profile exists and that the SAF-product is active. If a check fails indicating that any of these conditions is not met, access is granted, regardless. If this is not what you want, set the advanced automation option of AOF_AAO_SEC_PPIAUTH=FAIL.
4. Read “Step 14B: Install SA Provided Authorized TSO Command INGPAUTH” on page 104 to install INGPAUTH as an authorized TSO command. The TSO REXX functions use INGPAUTH under cover for RACF checking.

Resources

SA z/OS supports to secure resources to a certain degree.

If you want to limit or completely prevent that resources are manipulated then you need to turn on resource level security. This requires you to define profiles and permission statements. There are two types of profiles in the SYSAUTO general resource class.

1. Profiles for resources that are controlled by SA z/OS:

Syntax:

```
AGT.sysplexname.saxcfgroup.RES.resource_name.resource_type[.resource_location]
```

The variables have the following meanings:

sysplexname
This variable denotes the name of the physical sysplex.

saxcfgroup
This variable denotes the XCF group name for this particular system. The name always starts with the prefix INGXSG, followed by the value that is specified in the <sa_xcf_grpid_suffix> variable in the INGDOPT Configuration Options file.

resource_name
This is the name of the System Automation resource (for example, TSO).

resource_type
This parameter references the type of a resource (for example, APL, APG, MTR, SYG,...).

resource_location
This optional parameter references the location of a resource (for example, SYS1).

2. Profiles for SA z/OS special resources.

Syntax:

```
AGT.sysplexname.saxcfgroup.RES.special_res_name[.qualifiers]
```

The variables have the following meanings:

sysplexname
This variable denotes the name of the physical sysplex.

saxcfgroup
This variable denotes the XCF group name for this particular system. The name always starts with the prefix INGXSG, followed by the value that is specified in the <sa_xcf_grpid_suffix> variable in the INGDOPT Configuration Options file.

special_res_name
This is the name of a 'special' resource which is indicated by a leading underscore (for example, _MANAGER).

qualifiers
These are optional qualifiers for a special resource (for example, .DIAG).

Here are some examples for both kinds of resources:

AM Notation	SAF definition
Application TSO/APL/SYS1	AGT.SYSPLEX1.INGXSG.RES.TSO.APL.SYS1
Application Group BASE/APG/SYS1	AGT.SYSPLEX1.INGXSG.RES.BASE.APG.SYS1
Application Group AM_X/APG	AGT.SYSPLEX1.INGXSG.RES.AM_X.APG
Special Resource _CONFIG	AGT.SYSPLEX1.INGXSG.RES._CONFIG
Special Resource _MANAGER.DIAG	AGT.SYSPLEX1.INGXSG.RES._MANAGER.DIAG

You can use wildcards in the profile, when the class SYSAUTO has generics enabled. To enable generics, with RACF you can use the following command:

```
SETR GENERIC(SYSAUTO)
```

The affected resource(s) and authority required is determined by looking at the parameters and/or the panel input according to the following table:

Table 27. Resource and Profile Security Relationships

Resources	Profile	Command	Parameter
SA Resource (APL, APG, MTR, SYS, ...) Service period (SVP)	UPDATE	INGREQ	Unless CONTROL
		INGRUN	REQ=SET (affected SYG) REQ=ADD/DEL (what is added/ deleted)
		INGGROUP	RECYCLE, CANCEL, RESET, DEFAULT, EXCLUDE, AVOID, INCLUDE
		INGMOVE	Generally
		INGSET	CANCEL, KILL
		INGVOTE	CANCEL, KILL from full screen
		SETSTATE	Generally
		INGSUSPD	Unless CONTROL
		INGSCHED	REQ=DEL, REQ=REPL
		INGMDFY	Temporarily modify start or stop commands of a resource.
		INGVARS	Set, delete, or swap a shared variable that is associated with a resource.
	CONTROL	INGREQ	With SCOPE, OVERRIDE, INTERRUPT other than default (IBM supplied or installation)
		INGSET	SET
		INGGROUP	ACTIVATE, PACIFY, ADJUST
		INGSUSPD	With SCOPE other than default (IBM supplied or installation)

Table 27. Resource and Profile Security Relationships (continued)			
Resources	**Profile**	**Command**	**Parameter**
_MANAGER	UPDATE	INGAMS	SET
	CONTROL	INGAMS	DISABLE, ENABLE, SUSPEND, RESUME
_MANAGER.DIAG	UPDATE	INGAMS	DIAG REQ = other than STATS
_CONFIG	UPDATE	INGAMS	REFRESH
		INGCLEAN	Generally (note that security checks of ACF REQ=DEL are bypassed).
		ACF	REFRESH, ATLOAD
	CONTROL	ACF	COLD, REQ=DEL, REQ=REPL

For examples of how to define the profiles and permissions to secure resources, see the sample definitions in the INGESAF member.

Stylesheet Options

You learned about the role concept, definitions that are required for human operators, and how commands can be secured according to IBM recommendations. There are stylesheet options that are required to implement this level of security.

The CNMSTGEN member, generated by the Configuration Assistant, uses security options that start the Automation Agent before you do any configuration to this member, with the defaults that are provided by the product. However, this level of security is not sufficient and in fact is not secure at all, unless you change the default passwords as explained here.

When you are ready to switch to SAF-based security, in your <sa_hlq_user>.DSIPARM data set, edit the CNMSTGEN member and activate the following options:

The first option specifies that operator identification and password or password phrase checking is done with an SAF security product.

```
SECOPTS.OPERSEC = SAFDEF
```

The second option specifies that the NetView component performs command authorization checking with an SAF security product. Users can issue all commands when the SAF product cannot make a security decision. This option avoids the need to define profiles and permissions for all non-critical NetView component commands explicitly.

```
SECOPTS.CMDAUTH = SAF.PASS
```

The third option specifies to check the authority of the original issuer or the ID closest to the original issuer.

Make sure, you specify each of the options once and you comment out the default settings in this member.

```
SECOPTS.AUTHCHK = SOURCEID
```

The fourth option specifies that commands routed tasks from the NetView automation table are not authority-checked by a SAF security product, unless SEC=CH was specified on the CMDDEF statement.

```
DEFAULTS.AUTOSEC = BYPASS
```

You activate resource level security checks by setting the following stylesheet option in CNMSTGEN:

```
SECOPTS.SARESAUT = ON.PASS
```

The user id used for SAF checking is either OPID() from the top level System Automation command or explicitly set for third party checking (for example, from the PPI Receiver).

Restricting Access to Change PDB Activity Log Options

The activity logging of policy database (PDB) is protected by SAF. You can restrict who can modify PDB activity log options in the Customization Dialog, including turning activity logging on or off and changing the log data sets. This restriction ensures that all changes are logged into a controlled log data set.

To authorize a user to change the activity log options in the Customization Dialog, define a profile matching the following pattern in the SYSAUTO general resource class and give the user UPDATE access to the resource.

```
POLICY.enterprise-name.OPTIONS.ACTIVITYLOG
```

The actual enterprise name is used as the second qualifier. You can use wildcards in the profile, when the class SYSAUTO has generics enabled.

To restrict a user from modifying the activity log options, you need to define a profile for the user, but give him/her no UPDATE access.

For compatibility reason, if no such profile exists, the Customization Dialog treats the user as authorized to change the activity log options.

Other Security Options

Table 28 on page 168 shows you what other optional areas matter in terms of security. Also, where you can find detailed information for setting up your security correctly.

Table 28. Information References for Security

Area	Further Information
System Logger	See "Step 11: Configure the System Logger" on page 97 in Chapter 10, "Traditional SA z/OS Configuration," on page 67.
Joblog Monitoring	See "Access to JES Spool Output Data Sets" on page 172 and "Restricting Access to Joblog Monitoring Task INGTJLM" on page 174 of Chapter 11, "Security and Authorization," on page 151.
IPL Information	See "Access to IPL Information" on page 170 of Chapter 11, "Security and Authorization," on page 151.
Access to the NetView UNIX Command Server	See "Access to the NetView UNIX Command Server" on page 172 of Chapter 11, "Security and Authorization," on page 151.
Accessing authorized TSO command INGPAUTH	See "Step 14: Configure Function Packages for TSO" on page 103 of Chapter 10, "Traditional SA z/OS Configuration," on page 67 and also "Accessing authorized TSO command INGPAUTH" on page 172 of Chapter 11, "Security and Authorization," on page 151.

Table 28. Information References for Security (continued)	
Area	**Further Information**
Accessing the INGSUSPD suspend file	See “Accessing the INGSUSPD suspend file” on page 173 of Chapter 11, “Security and Authorization,” on page 151.
Security considerations to control Db2 subsys	See “Security considerations to control Db2 subsystems” on page 174 of Chapter 11, “Security and Authorization,” on page 151.
Requesting CEEDUMPs and DYNDUMPs	See “Requesting CEEDUMPs and DYNDUMPs” on page 174 of Chapter 11, “Security and Authorization,” on page 151.
Tivoli Monitoring	See “Security for IBM Tivoli Monitoring Products” on page 175 of Chapter 11, “Security and Authorization,” on page 151.
Processor Operations	See “Controlling Access to the Processor Hardware Functions” on page 180 of Chapter 11, “Security and Authorization,” on page 151.

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)
- 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 and password. 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 to the started task. User IDs are assigned either by using STARTED profiles or by using the ICHRIN03 table (see

z/OS Security Server RACF System Programmer's Guide). The user ID must have RACF UPDATE authority to the data sets. The data set names are created as follows:

```
hlq.domain.HSAyyddd.Xhhmmss
```

hlq
is the high-level qualifier for temporary data set defined during the configuration

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 to the started task. User IDs are assigned either by using STARTED profiles or by using the ICHRIN03 table (see *z/OS Security Server RACF System Programmer's Guide*). The user ID must have RACF UPDATE authority to the data sets. The data set names are created as follows:

```
hlq.domain.HSAyyddd.Xhhmmss
```

hlq
is the high-level qualifier for temporary data set defined during the configuration

domain
is the domain ID of the current NetView

X
O or P

Access to IPL Information

The automation function that collects, displays, compares, and deletes IPL information uses two started tasks. It is recommended that you 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 and the command can delete IPL information, both need RACF CONTROL access to the VSAM data set. The started task that collects 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) because this utility requires a fixed ddname. The input and output data sets that are used by the second started tasks are dynamically allocated and deleted by the NetView address space. This requires RACF ALTER access to these data sets for NetView. In case the security option OPERSEC is set to SAFDEF or SAFCHECK, the invoking NetView user ID must be granted with the RACF ALTER access. For any other OPERSEC values, the NetView started task user ID must be granted with this access.

When the address space of the started task is created, the operating system assigns a user ID to the started task (the IBM default is STCUSER). 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 sets are built as follows:

```
hlq.cdstype.Svvvvvv
```

Where:

hlq
is the high-level qualifier for couple data sets defined during the configuration

cdstype
is ARM, CFRM, LOGR, LOGRY, LOGRZ, SFM, SYSPLEX

Svvvvvv
is the volume name from the list of Alternate 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 sets are built as follows:

```
hlq.sysname.Vvolume.Snn
```

Where:

hlq
is the high-level qualifier for page data sets defined during the configuration

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

Access to JES Spool Output Data Sets

The task INGTJLM processes JES spool output data sets. It runs under the NetView user ID.

For this reason, the NetView user ID must be granted READ access to the class JESSPOOL in general or to those data sets in this class that will be monitored. The data set name of a JES spooled data set is built as follows:

```
localnodeid.uid.jobnm.jobid.dsidentifier.name
```

Where:

localnodeid
The NJE node name of the node on which SYSIN or SYSOUT data set currently resides. The *localnodeid* appears in the JES job log of every job.

uid
The user ID that owns the job.

jobnm
Job name.

jobid
The identifier of the job.

dsidentifier
It is the unique data set identifier that JES assigned to the spool data set. This identifier is 8 bytes of alphanumeric characters.

name
The name of the data set that is specified in the **DSN=** parameter of the DD statement. This name cannot be JESYSMSG, JESJCLIN, JESJCL, or JESMSGLG and follows the naming conventions for a temporary data set. If the JCL did not specify **DSN=** on the DD statement that creates the spool data set, JES uses a single question mark (?).

For more information, see "Defining profiles for SYSIN and SYSOUT data sets" in *z/OS Security Server RACF Security Administrator's Guide*.

Access to the NetView UNIX Command Server

If access to the NetView UNIX Command Server is required, it is necessary to define the <sa_hlq_smpe>.SINGLOAD library to PROGRAM CONTROL and permit the affected tasks appropriately:

```
RALTER PROGRAM ** +
   ADDMEM('<sa_hlq_smpe>.SINGLOAD'//NOPADCHK)
PERMIT ** CL(PROGRAM) ACCESS(READ) +
   ID(INGWRK)
SETROPTS WHEN(PROGRAM) REFRESH
```

Consult the INGESAF member generated by the Configuration Assistant.

For additional information on these commands, refer to *z/OS Security Server RACF Command Language Reference*.

Accessing authorized TSO command INGPAUTH

To secure the infrastructure of INGPAUTH, it is necessary to define the <sa_hlq_smpe>.SINGLOAD library to PROGRAM CONTROL:

```
RALTER PROGRAM ** +
   ADDMEM('<sa_hlq_smpe>.SINGLOAD'//NOPADCHK)
PERMIT ** CL(PROGRAM) ACCESS(READ) +
   ID(INGWRK)
SETROPTS WHEN(PROGRAM) REFRESH
```

Consult the INGESAF member generated by the Configuration Assistant.

For additional information on these commands, refer to *z/OS Security Server RACF Command Language Reference*.

Accessing the INGSUSPD suspend file

The suspend file is the data set containing the list of the suspended resources that is used by the INGSUSPD command. The list is maintained by a user ID that has an administrative role outside of the NetView environment. The user ID must at least have RACF UPDATE access.

The automation manager has the ability to modify the content of the suspend file. That's why the assigned started task user (the IBM default is STCUSER) running the automation manager must have RACF UPDATE access to the data sets.

Restricting Access to INGPLEX and INGCF Functions

This section describes how to control user access 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 INGPLEX or INGCF keyword, 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 to restrict access to the INGPLEX and INGCF functions:

Keyword	Value	Allows for
INGPLEX	CDS	• Allocating an alternate CDS with the INGPLEX CDS command • Controlling the SDUMP options and the SLIP traps sysplexwide
	HW	• Deactivating the LPAR of a CF with the INGCF DRAIN command • Activating the LPAR of a CF (equivalent to starting the Coupling Facility Control Code) with the INGCF ENABLE command • Including the INGCF keyword with the CF value
INGCF	CF	• Preparing to remove a CF from the sysplex with the INGCF DRAIN command • Integrating or reintegrating a CF into a sysplex with the INGCF ENABLE command • Including the INGCF keyword with the STR value • Including the INGPLEX keyword with the CDS value
	STR	• Forcing the deallocation of a CF structure with the INGCF STRUCTURE command • Rebuilding a CF structure on another CF with the INGCF STRUCTURE command • Controlling the SDUMP options and the SLIP traps sysplexwide

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.INGPLEX.HW
PROTECT *.*.INGRCCHK.INGCF.CF
PROTECT *.*.INGRCCHK.INGCF.STR
PERMIT GRP3 *.*.INGRCCHK.INGPLEX.CDS
PERMIT GRP3 *.*.INGRCCHK.INGCF.STR
PERMIT GRP4 *.*.INGRCCHK.INGCF.CF
PERMIT GRP5 *.*.INGRCCHK.INGPLEX.HW
```

With these definitions, operators of group GRP3 are authorized to issue all functions that require the authorization of INGPLEX=CDS or INGCF=STR.

Operators of group GRP4 are authorized to issue all functions that require INGCF=CF authority and all functions of GRP3, but are not authorized for the functions that require INGPLEX=HW authority.

Restricting Access to Joblog Monitoring Task INGTJLM

The task INGTJLM processes JES spool output data sets. It runs under the NetView user ID.

For this reason, the NetView user ID must have read access to the data sets being monitored. However, the permission allows all NetView users to read the spool data, even sensitive data, using the INGJLM command unless the command is restricted. Use the NetView command authorization table (see below) or the equivalent SAF command authorization to restrict the parameters START, STOP, and SUSPEND.

```
PROTECT *.*.INGJLM.START
PROTECT *.*.INGJLM.STOP
PROTECT *.*.INGJLM.SUSPEND
PERMIT grpx *.*.INGJLM.START
PERMIT grpx *.*.INGJLM.STOP
PERMIT grpx *.*.INGJLM.SUSPEND
```

Requesting CEEDUMPs and DYNDUMPs

Users running Language Environment applications or authorized key Language Environment applications must be permitted to request a CEEDUMP or a DYNDUMP.

For this reason, the userids of NetView started tasks and automation manager started tasks must have read access to resource profiles that belong to class FACILITY. Use the following definitions to grant access to these started task users (*stcuser*).

```
PERMIT IEAABD.DMPAUTH CL(FACILITY) ACCESS(READ) ID(stcuser)
PERMIT IEAABD.DMPAKEY CL(FACILITY) ACCESS(READ) ID(stcuser)
SETROPTS RACLIST(FACILITY) REFRESH
```

Security considerations to control Db2 subsystems

This section describes the necessary authorization that permits SA z/OS to control and to act on Db2 subsystems appropriately.

The SA z/OS provided Db2 best practices policy contains Db2 related commands to be issued as MVS command and a utility INGDB2. For more information about the INGDB2 utility, see the INGDB2 command in *IBM Z System Automation Operator's Commands*.

Granting SA z/OS to the Db2 SYSCTRL authority level enables SA z/OS to control the dedicated Db2 subsystem. The SYSCTRL authority is designed for administering a system that contains sensitive data. With the SYSCTRL authority, you have nearly complete control of the Db2 subsystem. However, you cannot access user data directly unless you are explicitly granted the privileges to do so.

If the Db2 is secured by RACF, use the following statements to permit SA z/OS to control the Db2 subsystem.

1. Permit SA z/OS to control a dedicated Db2 subsystem.

```
PERMIT <db2-subsystem>.SYSCTRL DSNADM CLASS(DSNADM) ID(AUTWRK01..) ACC(READ)
```

2. Permit SA z/OS to control all Db2 subsystems on a system.

```
PERMIT *.SYSCTRL DSNADM CLASS(DSNADM) ID(AUTWRK01..) ACC(READ)
```

If the Db2 security is located within the Db2 subsystem itself, then grant all SA z/OS work operators AUTWRK01 – AUTWRK*nn* to the Db2 SYSCTRL authority level.

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.

Please use the following RACF instructions to add the certificates uploaded in “Step 38A: Enabling SOAP over HTTPS for a TEMS” on page 148 to the user's keyring.

```
racdcert id(#saf_user#) addring(<keyring>)
racdcert id(#saf_user#) add ('<UID.ITM.PEM>') WITHLABEL ('ITM') TRUST
racdcert id(#saf_user#) connect (ID(#saf_user#) RING(<keyring>) LABEL('ITM') USAGE(CERTAUTH)
setropts raclist (digtring) refresh
setropts raclist (digtcert) refresh
```

#saf_user#
represents the userid authorized for these certificates. If NetView option SECOPTS.OPERSEC is set to SAFDEF, then each human and ISQ* operator must be authorized separately. Otherwise the started task must be authorized.

For more information, refer to “Step 38A: Enabling SOAP over HTTPS for a TEMS” on page 148 for the SSL socket connection.

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, see the appendix "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 29 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 INITIAL0 rule. In the absence of INITIAL*n* rules, the command authority level for SA z/OS is always 0. For further details, see 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, see 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 configuration that is required in OMEGAMON to permit access to the monitor.

Table 29 on page 176 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 29. Command Authorization Identifiers

Commands and Keywords	**Command List Name**	**SAF Resource or Command Authorization Table Identifier**
INGOMX NAME CMD SERVER IPADDR	INGROMX0	*netid*.*luname*.INGROMX0 *netid*.*luname*.INGROMX0.NAME.*session_name* *netid*.*luname*.INGROMX0.CMD.*command* *netid*.*luname*.INGROMX0.SERVER.*server_name* *netid*.*luname*.INGROMX0.IPADDR
INGSESS REQ START STOP	INGRYSS0	*netid*.*luname*.INGRYSS0 *netid*.*luname*.INGRYSS0.REQ *netid*.*luname*.INGRYSS0.REQ.START *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 or to the SOAP server requires authentication with a user ID and password if product level security is active. For a OMEGAMON session, the user ID and password are passed from the OMEGAMON Session Attributes policy. For a SOAP request, the user ID and password are passed from the SOAP SERVER policy, or passed from the INGOMX command if the SOAP server is not defined in the policy.

In the OMEGAMON Session Attributes and SOAP SERVER policy items (see the following panel as an example), you can specify the password in clear text or specify the keyword SAFPW or PTKT.

```
  COMMANDS  HELP
------------------------------------------------------------------------------
AOFGOSOA                   OMEGAMON Session Attributes
Command ===> ____________________________________________________

Entry Type : Network               PolicyDB Name   : USER_PDB
Entry Name : ALL_SYSTEMS_NETWORK   Enterprise Name : USER_ENTERPRISE

Session Name   : SESS1                                                    .
...
OMEGAMON type .  OMIIIMS     (OMIICICS OMIIDB2 OMIIIMS OMIIMVS)
SAF applid . . . CANDLE      SAF application ID for OMEGAMON
User ID. . . . . SAIMS
                             User ID to log on to OMEGAMON
Password . . . . SAFPW       Password of the logon user or keyword SAFPW or PTKT
...
```

Figure 11. Password specification in the policy

- Specify the password in clear text

  When a password is specified in the OMEGAMON Session Attributes or SOAP SERVER policy items, it appears in readable format in the automation configuration file and in logs.

- Specify the keyword SAFPW to use the password stored in a data set

  When SAFPW is specified, the password is stored in a VSAM data set in an encrypted format. The SA z/OS command INGPW is used to access the password data set to set or read the password.

  For more information, see “Authentication Using the SA z/OS Password Data Set” on page 178.

- Specify the keyword PTKT to use RACF PassTicket (OA64126)

  When the predefined value PTKT is specified, SA z/OS can obtain a RACF PassTicket, which is a one-time-only password for the user ID. To use this keyword for SOAP requests, the Hub TEMS must run on z/OS. The Hub TEMS and OMEGAMON authentication must be configured for SAF usage.

  For more information, see “Authentication using PassTickets” on page 178.

Authentication Using the SA z/OS Password Data Set

The SA z/OS password data set is used as a password safe if you do not want to reveal passwords in your policy database.

The SA z/OS command INGPW is used to access the password data set to set or read the password. SA z/OS uses INGPW as follows:

- Passwords are stored and retrieved by *user_id* and *owner_id*.
- *user_id* is the common user defined to log on to an OMEGAMON session.
- *owner_id* 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.

See INGPW command in *IBM Z System Automation Operator's Commands* for further details.

Procedure

To use SA z/OS Password Data Set, complete the following setup activities.

1. The password data set has to be created first and allocated upon the start of NetView. See “Step 2D: SA z/OS Password Store Data Set” on page 72 for details.
2. Set the initial password for a user ID with a given owner in the password data set using the SA z/OS command INGPW.

   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, INGPW 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 INGPW. 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 INGPW command to initialize the password data set. For example, suppose that the session and password share definitions are set for user `oper1` and owner `AOMON`, the INGPW command format would be:

   ```
   INGPW oper1 AOMON,INIT=pw,MASK=%A%N%N%A%A%A%A%A,EXPINT=0
   ```

   pw is the initial password for the user ID. The MASK parameter indicates that the password should be 8 characters long, beginning with a letter, followed by 2 numbers and then 5 letters and never expire.
3. All applications denoted by the OMEGAMON applied that share the same password must be assigned to a single owner. You define the owner in the NETWORK (NTW) entry type with the AUTHENTICATION policy item. On the Authentication Definitions panel, enter your definitions in the **Owner** and **Share** fields. See "AUTHENTICATION Policy Item" in *IBM Z System Automation Defining Automation Policy* for more details about this panel.

Authentication using PassTickets

You can use RACF PassTickets to authenticate the user that attempts to log in to OMEGAMON or make a SOAP request. This feature avoids storing password credentials in clear or managing the passwords by using the System Automation's password data set. It's an enhancement that is delivered with APAR OA64126.

The RACF PassTicket is a one-time-only password that can be generated by System Automation. For more information, see The RACF PassTicket in z/OS documentation.

Prerequisite

- To use PassTickets for OMEGAMON log-ins, OMEGAMON authentication must be configured for SAF usage.
- To use PassTickets for SOAP requests, the Hub TEMS must run on z/OS and Hub TEMS authentication must be configured for SAF usage.

Procedure

To use the z/OS PassTickets function, complete the following setup activities.

1. Activate the security class PTKTDATA. Optionally, if you want to use generic profiles, specify the **GENERIC** option with the command.

   ```
   SETROPTS CLASSACT(PTKTDATA) RACLIST(PTKTDATA) GENERIC(PTKTDATA)
   ```

2. OMEGAMON security authenticates a user for a given SAF application name, which defaults to CANDLE. Create an application profile for the SAF application in class PTKTDATA. Specify the PassTicket key in the profile. For example, you can use the following approach:

   ```
   RDEFINE PTKTDATA CANDLE SSIGNON(EPTKEYLABEL(MY.HMAC.KEY))
   ```

3. Grant access to appropriate users to call the INGOMX command.
   - To use PassTicket for OMEGAMON session authentication, the INGOMX caller must have READ access to the following resource in resource class SYSAUTO.

     ```
     AGT.sysplexname.saxcfgrp.RES._PASSTICKET.INGOMX.CLASSIC.userid
     ```

   - To use PassTicket for SOAP request authentication, the INGOMX caller must have READ access to the following resource in resource class SYSAUTO.

     ```
     AGT.sysplexname.saxcfgrp.RES._PASSTICKET.INGOMX.SOAP.userid
     ```

4. For the settings to become active, refresh the PTKTDATA and SYSAUTO classes in the RACF settings by using the **SETROPTS** command.

   ```
   SETROPTS RACLIST(PTKTDATA) REFRESH
   ```

   ```
   SETROPTS RACLIST(SYSAUTO) REFRESH
   ```

5. In addition, set the NetView SECOPTS.OPERSEC in the CNMSTGEN member to either option SAFCHECK or SAFDEF. This setting enables granular checking whether particular users or operators are authorized to create a PassTicket for themselves or anyone else.

After all updates are made in RACF and the CNMSTGEN member, PassTickets can be generated and evaluated by System Automation. Every target user that needs to work with the INGOMX command can use the PassTicket to authenticate their user IDs.

If authentication failed, ING168I message is issued in the syslog.

`CALLER IS NOT AUTHORIZED TO GENERATE PASSTICKET REASON: reason`

If the system was not able to generate a PassTicket, ING169I message is issued in the syslog.

`ING169I UNABLE TO GENERATE PASSTICKET REASON: rr`

For more information about ING168I and ING169I messages, see the *IBM Z System Automation Messages and Codes* manual.

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, an 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 NetView with SA z/OS can use the internal interface, the related application resources must be defined to the security access facility (SAF) of the system.

About this task

There are two application resources names (*app_res*):

HSAET32
For the INTERNAL connection protocol used by GDPS and the integrated sysplex automation functions of SA z/OS.

ISQET32
For the Processor Operations function of SA z/OS and its internal interface based hybrid SNMP connection protocol.

Depending on your SA z/OS customization, either one or both the application resource names must be defined.

Procedure

1. Prepare an SAF security class.
2. Define resource HSA.ET32OAN.*app_res* in the CLASS FACILITY.
3. Grant NetView READ access to this facility class resource.

Results

The following example shows the RACF commands used to prepare the access class, to define the application resource, and to grant the required READ access for the NetView user to the application resource.

```
SETROPTS CLASSACT(FACILITY)
SETROPTS RACLIST(FACILITY)
RDEFINE FACILITY HSA.ET32OAN.app_res UACC(NONE)
PERMIT HSA.ET32OAN.app_res 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 **RDEFINE** command fully qualifies the *app_res* 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.

Depending on your SA z/OS customization, the **RDEFINE** and **PERMIT** commands must be repeated with the second *app_res* name.

Access to the CPCs

Each processor (CPC) defined in your SA z/OS policy database 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
- UPDATE: 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 (OPERSEC) chosen, the access level is checked differently.

If your NetView operator security is set to 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 hardware interface autotasks AUTHW*nnn*
- Any operator issuing a hardware 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 hardware interface initialization and communication also need to be authorized. Use access level CONTROL for the AUTHWnnn 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.

Password Management for SNMPv3 HMC/SE Connections

Connecting to a HMC or SE using SNMP Version 3 protocol requires authentication with predefined SNMPv3 user name and password. Note that when a password is specified in the SA z/OS PROCESSOR INFO policy, it appears in readable format in the automation configuration file and in logs.

When SAFPW is specified, the password is stored in a VSAM data set in an encrypted format. You define the SNMPv3 user name and password on the HMC or SE (see “Step 7A: Preparing the HMC” on page 85 and “Step 7B: Preparing the SE” on page 87) and specify them in the PROCESSOR INFO policy item for processors using the SNMP connection protocol. Use the predefined value SAFPW to allow NetView to maintain the password of the user name. See "PROCESSOR policy item" in *IBM Z System Automation Defining Automation Policy*.

The SA z/OS command INGPW is used to access the data set to set or read the password. SA z/OS uses INGPW as follows:

- Passwords are stored and retrieved by *user_id* and *owner_id*
- user_id is the SNMPv3 user name defined for CPC
- owner_id is the ProcOps Target HW Name of the CPC as used by the SA z/OS customization dialogs for the CPC.

Authentication using the SA z/OS Password Data Set: The SA z/OS 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. See “Step 2D: SA z/OS Password Store Data Set” on page 72, for further details.

You are responsible for setting the password for a SNMPv3 user name with a given owner in the password data set using the SA z/OS command INGPW. The SNMPv3 password must be 8-31 characters long in order to be used with INGPW.

Whenever an SNMP session is established to SE or HMC, 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, INGPW returns the current password, otherwise the connection fails. You are responsible for ensuring that the password in the password data set and password known to the SE or HMC are the same, in particular if you plan to use an alternate focal point. In this case, the password data sets should be shared by the group of systems where the focal point can run.

Use the INGPW command to initialize the password data set. For example:

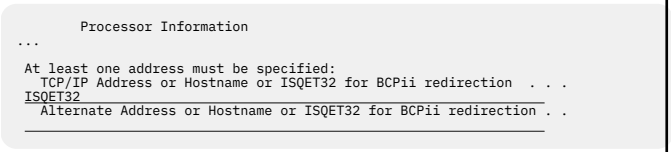
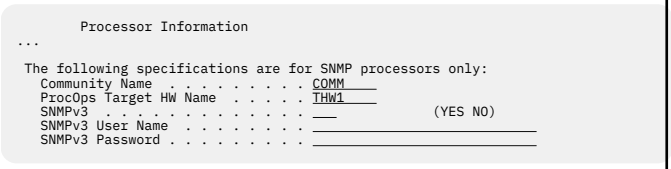
```
NETVASIS INGPW v3testuser T99PRO, INIT=SNMPV3PWD,MASK=%A%A%A%A%A%A%A%A,EXPINT=0
```

Note: System z API does not support remote SNMPv3 password change, therefore do not use automatic password expiration feature of the INGPW command (use EXPINT=0 or omit the parameter). See *IBM Z System Automation Operator's Commands* for further details about the INGPW command.

Processor Hardware Connection Security Considerations

IBM Z System Automation supports TCP/IP and SA-BCPii as the transport protocols for Support Element connections and TCP/IP for Hardware Management Console connections.

Inside IBM Z System Automation, TLS itself is not yet supported for any hardware automation connection. But you can use the hybrid SNMP connection (ISQET32) or the INTERNAL connection. Since ISQET32 or INTERNAL communication is kept inside IBM Z hardware, which includes the IBM Z hardware network, there is no need for additional transport security such as TLS to secure it in a public network environment. Ensure that you're aware of the following security considerations:

Connection Type	Processor Type	
	Mainframe	**ProcOps Service Machines (PSM)**
Hybrid SNMP	• For ProcOps SNMP connections, only configure ISQET32 (hybrid SNMP) as the hostname. Do not define an HMC IP address or hostname.  • The new **SNMPv3 TLS configuration** option in the **Customize API Settings** task, which is introduced with IBM z16, is not supported by System Automation. • The current SNMPv3 support in ProcOps is not TLS-compliant. 	This connection type is not valid for PSM.
INTERNAL	• INTERNAL connections are SA-BCPii connections to the processor's Support Element. • SA-BCPii connections do not require TLS level security. The connection end points are inside the IBM Z machines. The IBM Z CPC network used between multiple IBM Z systems is isolated from any customer network.	This connection type is not valid for PSM.
TCP/IP	This connection type is not valid for the mainframe type of processor.	TCP/IP is the only valid connection option for PSM. But the underlying socket services do not exploit TLS.

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*.

Chapter 12. Configuring SA z/OS Workstation Components

This information contains information about how to install those parts of SA z/OS that are required on workstations:

- "Configuring IBM Tivoli Netcool/OMNIbus" on page 185
- "Configuring Tivoli Service Request Manager through Tivoli Directory Integrator" on page 187

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 that are being managed with SA z/OS.

Configuring IBM Tivoli Netcool/OMNIbus

Because SA z/OS uses Tivoli Event Integration Facility (EIF) events for communication you need the following components:

About this task

- IBM Tivoli Netcool/OMNIbus (OMNIbus)
- The OMNIbus Probes Library for Nonnative Base
- The Tivoli EIF Probe (EIF Probe)

It is assumed that you have all of the above installed and verified before you begin with the customization for SA z/OS. For details please see the product manuals.

For more information about the infrastructure on host systems, refer to "Step 15: Configure Alert Notification for SA z/OS" on page 104.

Although OMNIbus can run on various operating systems the following example describes the installation and customization on Windows 2003 Server.

Procedure

1. Download the sample files `ING_event.rules` and `ING_db_update.sql` from the host system to your workstation as text files:
   a) To download the files, you can use, for example, FTP. Choose as the target path name any directory where you want to store temporarily the sample files:

   ```
   cd <PATH>
   ```

   b) Start FTP with:

   ```
   ftp <hostname>
   ```

   c) You will be prompted to enter your user ID and password. After logging on to your z/OS system, enter:

   ```
   ascii
   get /usr/lpp/ing/dist/OMNIbus/ING_event.rules
   get /usr/lpp/ing/dist/OMNIbus/ING_db_update.sql
   quit
   ```

2. Inspect `ING_db_update.sql`. This file creates new columns in your ObjectServer's alert.status table that will later hold the information from the SA z/OS events. It will also add some triggers and a trigger group. Normally you should not have to change this file.
3. Update the alert.status table of your ObjectServers:

a) Run the SQL processor:

```
%OMNIHOME%\bin\redist\isql.exe -S <server> -U <username>
-P <password> -i <PATH>/ING_db_update.sql
```

b) Repeat the previous step for each ObjectServer.

4. Adapt your EIF probe `tivoli_eif.rules`. There are two possibilities:
   - Your Tivoli EIR Probe is for SA z/OS events only so you can simply replace the original rules file with the one supplied by SA z/OS:`copy ING_event.rules C:\Program Files\IBM\Tivoli\Netcool\omnibus\probes\win32\tivoli_eif.rules`
   - Otherwise you must merge the logic of `ING_event.rules` into your existing `tivoli_eif.rules`
5. Restart your ObjectServers and your EIF Probe.

Configuring the Triggers

`ING_db_update.sql` installs a trigger called `ing_count_events`. This trigger is designed to prevent multiple lines to be displayed for multiple occurrences of the same event.

About this task

Instead of that it maintains a counter that is increased each time the same event arrives repeatedly. The `ing_count_events` trigger is initially disabled because the installation process of the EIF Probe installs another trigger called deduplication. If you have both triggers enabled your event counter will be increased twice.

You should proceed based on the following options:

- Your EIF Probe is for SA z/OS events only: It is recommended that you have `ing_count_events` enabled and deduplication disabled.
- Your EIF Probe is also for other events: You must review both triggers and merge the logic.
- You want to see all occurrences of an event as a separate line: You must disable both triggers.

Note that you can manipulate the triggers in IBM Tivoli Netcool/OMNIbus Administrator by connecting to your ObjectServers and selecting **Automation > Triggers**.

Configuring the Event View

The event views of IBM Tivoli Netcool/OMNIbus Conductor can be customized to show the fields that have been newly inserted into the alert.status table for SA z/OS events.

About this task

In the event view select **Edit > Edit View**.

A recommended setup is:

- Node
- AlertGroup
- Summary
- Tally
- INGEventDate
- INGEventTime
- INGEventResName
- INGEventResType
- INGEventResSystem
- INGEventJobname

Note: SA z/OS uses the OMNIbus event class 89320. Make sure that you define this class.

Configuring Tivoli Service Request Manager through Tivoli Directory Integrator

About this task

Because SA z/OS integrates with IBM Tivoli Service Request Manager (TSRM) through IBM Tivoli Directory Integrator (TDI) you need the following components:

- TSRM and all prerequisite software
- TDI Runtime Server and Config Editor

It is assumed that you have all of the above installed and verified before you begin with the customization for SA z/OS. For details see the product manuals.

To create a trouble ticket from SA z/OS in TSRM there are no adaptations required in TSRM. Everything is done in TDI. Although TDI can run on various operating systems the following example describes the installation and customization on Windows 2003 Server.

Procedure

Download the sample file `ING_event.xml` from the host system to your workstation as a text file:

a) To download the file, you can use, for example, FTP. Choose as the target path name any directory where you want to store temporarily the sample files:

```
cd <PATH>
```

b) Start FTP with:

```
ftp <hostname>
```

c) You will be prompted to enter your user ID and password. After logging on to your z/OS system, enter:

```
ascii
get /usr/lpp/ing/dist/TDI/ING_event.xml
quit
```

Configuring the AssemblyLines

About this task

To perform the steps described in this section you should be familiar with the TDI Config Editor. A good overview can be found in *IBM Tivoli Directory Integrator User's Guide*.

The sample file `ING_event.xml` defines two AssemblyLines:

- TicketServer that receives a request from SA z/OS, starts TicketWriter and returns a response
- TicketWriter that parses the request and creates a trouble ticket in TSRM

Note that if you have a different service desk than TSRM you can adapt TicketWriter to feed your application. TicketServer can remain the same.

Because they are samples, the AssemblyLines will probably not work unchanged in your environment. You should review both and make any necessary adaptations:

Procedure

1. Start the TDI Config Editor and open <PATH>`ING_event.xml`.

2. Modify the AssemblyLine TicketServer as follows:
   a) Open TicketServer and select the **Data Flow** tab.
   b) Open the ReadXML component in the **Feeds** section.
   c) Adapt the port number. This is a TCP Connector working in server mode. The **Config** tab shows the port number that the server listens to. A value of 8000 is provided in the sample but you are free to change it.
   d) Leave the other components unchanged.
   e) Start the Ticketserver
3. Modify the AssemblyLine TicketWriter:
   a) Open TicketWriter and select the **Data Flow** tab.
   b) Modify how the details text is generated:
      i) Review all of the components with names like `Map...Description`.
      ii) The FixDescription and SpecificDescription attributes are set to text that is formatted with the attributes that are mapped by the `Map...Attributes` components. You can adapt the text your needs here.
   c) Modify the TSRM settings:
      i) Open the WriteTicket component. This is a Generic Maximo Connector.
      ii) Adapt the TSRM communication settings. Select the **Config** tab. Specify various options that must match your TSRM installation:

Configuration Tab	Action
MEA Server	specify the URL (server address port) of your TSRM
MEA Objects	specify a setting such as the external system name and the names of the Web services for CREATE, DELETE, QUERY and UPDATE operations
MEA Advanced	leave as is.

   d) On the **Output Map** tab TicketWriter sample maps DESCRIPTION and DESCRIPTION_LONG DESCRIPTION, as well as REPORTEDPRIORITY, URGENCY and IMPACT. You can also use this tab to map fixed installation-dependent values.

      The sample maps the REPORTEDBY user ID to the value SAZOS. You may want to change this or add other user IDs, or do both.

Appendix A. Using the Hardware Integrated Console of IBM Z for External Automation with SA z/OS

The Hardware Integrated Console provides a message and command interface for operating system images running on IBM Z hardware to cover system initialization, recovery situations, or emergency operator tasks.

Especially when channel-attached or otherwise-connected 3270/ASCII operator console devices are not configured or cannot be used with the IBM Z processor hardware, the integrated console is the only console interface for an operating system at initialization time.

For the SA z/OS processor hardware interfaces, the integrated console is the exclusive facility to communicate with the target operating systems running on IBM Z processors. Other console interfaces that become available after target OS initialization is complete are not used. With the SA z/OS hardware interfaces, you can control and automate IBM Z processors externally. This means the controlling SA z/OS program can run on a different processor or LPAR than the target system to be controlled. One typical example is to monitor or automate the IPL prompts of a remote system displayed on its integrated console.

This appendix provides background, usage, and performance information important to know if you plan to use the hardware integrated console support (CI) of the SA z/OS processor hardware interfaces for your automation. The IBM Z hardware commands, like SYSRESET, LOAD for example, are not discussed in this chapter. For more information about automating these commands, refer to *IBM Z System Automation Operator's Commands* and *IBM Z System Automation User's Guide*. However the automation interface and remote configuration information in this chapter is valid for both hardware commands and CI automation. This appendix includes the following sections:

How HMC Integrated Console Tasks impact System Console Message Automation

Note functional compatibility issues are described in these sections that are related to IBM Z hardware and IBM Z firmware. SA z/OS as an HMC/SE function exploiter and IBM Z SNMP APIs cannot bypass or circumvent the mentioned automation impacts in its product code or documentation. z/OS is responsible for its implementation of 'Integrated 3270 Console' support as HMCS console.

The HMC offers two different Console Message Interfaces

As a HMC user, you can use the recovery tasks 'Integrated 3270 Console' or 'Integrated ASCII Console' to work with a console emulation session as an alternative to the 'Operating System Messages' HMC/SE window to monitor operating system console messages or to issue commands to an operating system running in a CPC partition. Since the 'Integrated Consoles' have the look and feel of screen emulation sessions, rather than being a message box with limited console functionality, it is likely that using the 'Integrated Console' becomes more common than using the 'Operating System Messages' window. From a SA z/OS hardware interface perspective, you should be aware of the side effects, the 'Integrated Console' usage can have for the console message based automation of SA-BCPii and ProcOps. Both protocols use the IBM Z SNMP API function which allows you to wait for operating system message events, emitted in the operating system (OS) message window. For SA-BCPii and ProcOps this is the unique source for OS console message automation. There is no IBM Z SNMP API OS message interface to 'Integrated Consoles'.

Initial z/OS IPL Messages

Introduced with z/OS 2.1 and not requiring another configuration step, a detected active HMC 'Integrated 3270 Console' session assigned to the LPAR being loaded, causes all initial messages (z/OS NIP messages) to be sent to the 'Integrated Console', but no longer to the System Console, although the IOCD NIPCONS definitions may have been set up in this way. As a result, no z/OS NIP reply prompt or action message is routed to the System Console message window to be displayed. Implicitly, this prohibits any z/OS NIP message automation from SA-BCPii or ProcOps.

z/OS console messages

After the z/OS IPL NIP phase, when the CONSOLxx definitions come into effect, the System Console may again show z/OS messages, which then can trigger SA-BCPii or ProcOps console message automation.

Other Operating Systems (OS) or Stand Alone (SAL) Utility Messages

Depending on the OS type or SAL utility, different configuration settings may be necessary to control the System Console usage or the ability to exploit the 'Integrated Console' function of the HMC. You should be aware of the impacts, illustrated here for z/OS. Refer to the appropriate OS and SAL documentation for more information about the IBM Z HMC 'Integrated Console' or Operating System Message window usage.

Monitoring and controlling 'Integrated Console' usage

The usage of this HMC task cannot be monitored with SA z/OS. Currently there is no way for SA-BCPii or ProcOps sessions to determine if there is an 'Integrated Console' session active. Neither security log entries nor HW messages provide information about this task invocation. No SE or HMC check-box prompts you in the case of a manual LOAD, that an active 'Integrated Console' may be disruptive for automated IPLs. There is also no IBM Z SNMP API flag indicating this. In addition, there is no way on the HMC to block this in general or for selected CPC partitions. Note, that just 'disconnecting' as an HMC user, keeps user started 'Integrated Console' sessions always active.

Avoid using the 'Integrated Console'

If you have system environments that use and depend on automated IPLs, do not use the 'Integrated Console' function at all. If that is not possible, you may at least reduce the IPL message automation impact risk by implementing a HMC user-based function limitation. Allow only a few users to use the 'Integrated Console' task for a limited number of LPARs. This does not generally eliminate the risk of outages due to missing IPL messages on the System Console, but can help to lower it. Refer to the IBM Z HMC manuals about HMC security and user roles, available at IBM Resource Link or download on the HMC Web client.

CI Usage in IBM System Automation Products

SA z/OS Processor Operations (ProcOps)

Processor operations is a NetView based automation interface and API to monitor and control IBM Z mainframes.The function uses the SNMP-TCP/IP and BCPii communication protocols. ProcOps is a focal point application that allows external mainframe automation. See the ProcOps API command ISQSEND in *IBM Z System Automation Operator's Commands* as an example of a ProcOps command using CI. The integrated IPL automation for z/OS and z/VM are other examples of using CI. With ProcOps, CI messages are sent automatically to the focal point system as soon as the network connection is established to the Support Element (SE) or Hardware Management Console (HMC) and the targeted system (LPAR) is registered. The ProcOps API command ISQXIII is used to perform these steps.

Related Information

The IBM Service Offering GDPS, an IBM disaster recovery solution for IBM Z mainframes, requires NetView and SA z/OS to be active. It uses the CI facility with the internal services of SA z/OS. Refer to the *GDPS Metro Installation and Customization Guide* for more information. An example of CI exploitation of GDPS is DUPLICATE VOLSER automation at IPL time.

Depending on the function performed, CI message registration or deregistration is controlled internally by the GDPS code.

CI Protocols and Automation Interfaces

In order to use the hardware integrated console (CI), the SA z/OS program uses two communication protocols. These protocols use the IBM Z SNMP application programming interfaces.

You use Option 10 (Processors) on the Entry Type Selection panel of the SA z/OS customization dialog to configure the communication protocols for a processor. See "Processor Entry Type" in *IBM Z System Automation Defining Automation Policy* for more information.

INTERNAL (BCPii Base Control Program Internal Interface)

This protocol is based on an IBM Z internal communication service (SCLP) between the LPARs and the processor support element (SE) to perform hardware operations and configuration management tasks. No network IP stack is needed. See “Planning the Hardware Interfaces” on page 22 for more information. The scope of processors that can be controlled with this protocol is the Hardware LAN.

SNMP

This protocol requires a Internet Protocol network stack. From a ProcOps focal point system, which must be connected to a business LAN, you can monitor and control processors and operating system messages (CI) from LPARs running on the controlled processors. Network access from the business LAN to the hardware LANs of the processors is required. ProcOps supports SNMP connections to HMCs and SEs.

Note: This is a hybrid interface, allowing you to redirect communications over BCPii instead of TCP/IP. Use hostname ISQET32 as the SNMP IP address for the SE or HMC in the SA PDB processor policy to define BCPii redirection.

IBM Z SNMP Application Programming Interface

The API covers all network-specific programming services (Bind, Connect, and so on) and allows applications to concentrate on hardware function and event control. The API uses the SNMP MIB data format. Applications using the API can dynamically register for events, such as operating system messages, from the CI of a particular LPAR.

For detailed information, refer to *IBM Z SNMP Application Programming Interfaces*, which is available under your HMC's **Books View** or on IBM Resource Link® for download. The document also contains information about how to download the API itself for various OS platforms and Java. This generally available API version supports the TCP/IP SNMP protocol.

A special version of the API is distributed with the SA z/OS that supports the BCPii and the TCP/IP protocol. This version can only be used together with SA z/OS.

Related Information

With z/OS, another BCPii can also be used independently of SA z/OS or GDPS by applications that are written in high-level languages to automate CI operations. See *z/OS MVS Programming: Callable Services for High-Level Languages* for more information.

CI Configuration for Remote Automation

Figure 12 on page 193 illustrates how the CI of three systems is connected to an SA z/OS system, which is acting as a remote automation focal point.

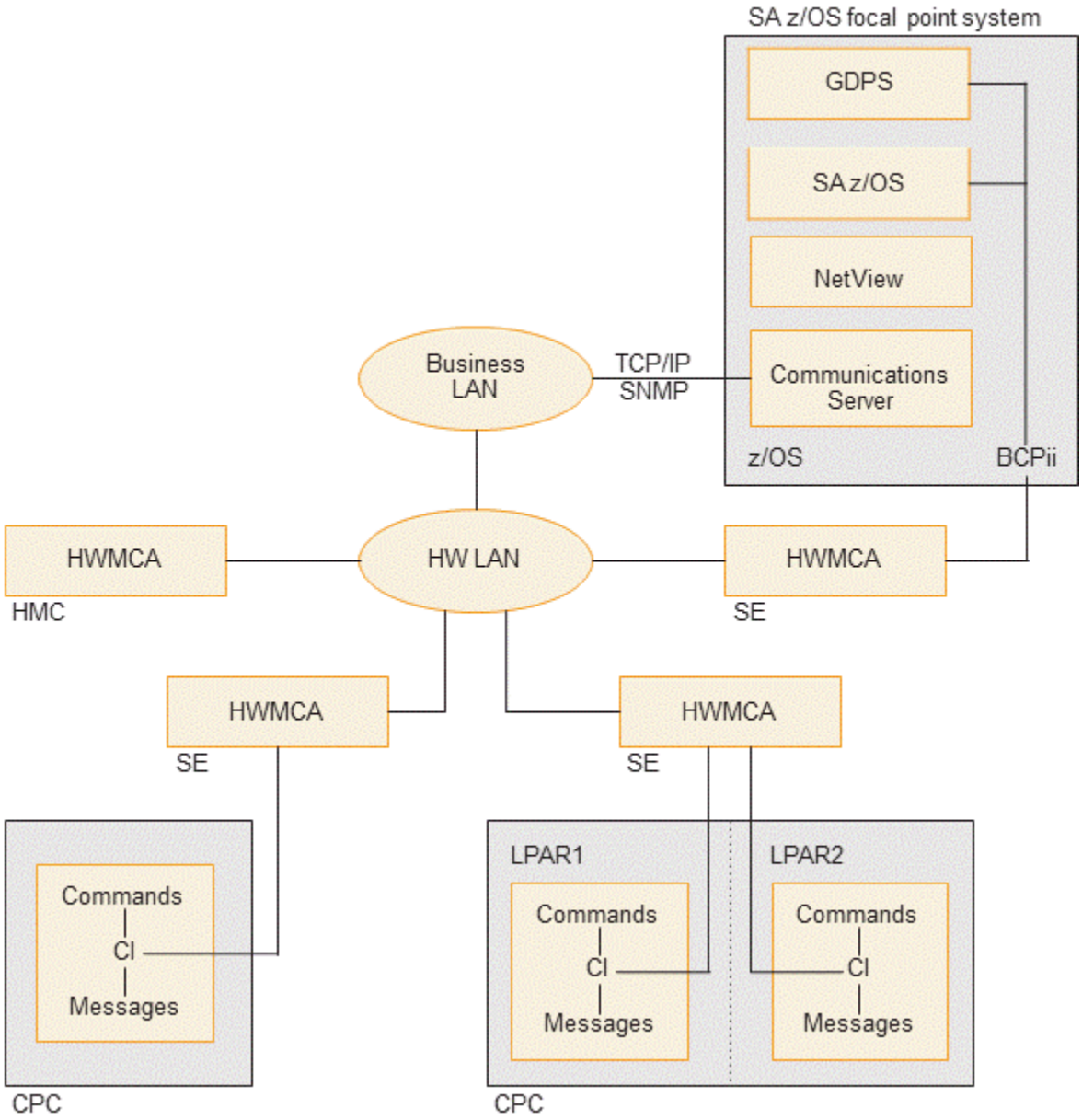


Figure 12. Remote Operations Components for IBM Z

SA z/OS ProcOps uses the TCP/IP connections that are always from a focal point (ProcOps FP System) to target processors and systems. The SA z/OS BCPii (INTERNAL) is a peer connection protocol. In a system cluster like a z/OS Parallel Sysplex, all participating systems can be configured in the SA z/OS policy to have BCPii connections with one another.

Focal points can be located close to the systems they control or located remotely from them. For the TCP/IP SNMP protocol that is used by SA z/OS this can be a Business LAN or Intranet, or a global Internet distance. For the BCPii (INTERNAL) protocol the distance between two BCPii connected systems depends on the dimension of the Hardware LAN.

With GDPS in a Parallel Sysplex environment, the distances between BCPii-connected systems is also affected by the connectivity requirements of the Coupling Links. Refer to the IBM Redbooks® publication, *IBM Z Connectivity Handbook* and the available GDPS documentation for more information.

How the Hardware LAN is connected to the Business LAN depends on the security policies that apply. Router/Bridge hardware and firewall software are typically used to control access. For more information refer to the *Installation Manual for Physical Planning* and *System Overview* manuals that are available for your IBM Z mainframe.

The Hardware Management Console Application (HWMCA) is a licensed software application that is installed on the Hardware Management Console and the Support Element (SE). It provides the GUI and the interfaces for automation software. BCPii connections and TCP/IP SNMP connections use the HWMCA.

SA z/OS ProcOps runs as a NetView application and uses a Communications Server TCP/IP stack to communicate with an SE or HMC. In Figure 12 on page 193, the HMC is attached to the Hardware LAN of the mainframes, however configurations with HMCs that are attached to the Business customer LAN are also supported. Support Elements must be attached to an HareWare LAN. CI message events and commands are exchanged between the connection end points of the SE or HMC and the SA z/OS ProcOps application.

GDPS, which runs as a NetView application, uses SA z/OS internal services to communicate with the Support Elements over the BCPii. The BCPii protocol itself uses the z/OS support processor interface services (SCLP) to do this. If a GDPS BCPii request targets an SE other than the local one, the HMC is used to route the request to the target.

In Figure 12 on page 193, the CI of three target systems is shown. One CPC has two logical partitions, LPAR1 and LPAR2, each with a CI. The third CI is shown for a single system that is running on another CPC. Together with the CPC of the focal point system, all the CPCs are connected to the same Hardware LAN.

Although not shown in Figure 12 on page 193, a fourth CI, that of the focal point system itself, can also be automated. Both of the TCP/IP SNMP and BCPii protocols can be used to do this.

CI Automation Basics

The CI facility uses a physical (cable) connection between the processor hardware (CPC) and the attached processor support element (SE) unit. With the CI, the message and command information is exchanged between a system image running on the CPC and its SE.

For automated operations, CI has an interface to the console application (HWMCA), running on each SE or HMC. If there is a ProcOps session to a HMC/SE, or a GDPS session to a SE, the console application generates an event for each new CI message. This event is sent to all registered applications (ProcOps, GDPS), using the transport protocol configured in SA z/OS. This is SNMP for ProcOps or INTERNAL (BCPii) for GDPS.

Automation applications can send operating system console commands to a CI for execution. With SA z/OS this can happen either in response to messages that are received only over CI, or independent of that at any time. The only requirement is that a SA z/OS hardware session exists between the SE/HMC and the automation application (SA z/OS/ProcOps or GDPS). The advantage for automation of using the CI is that there is no 3270-specific information and screen formatting burden. This makes the interface robust and easier to use for automation purposes than 3270 console screen emulation and interpretation.

Related Information

The Support Element (SE) provides the GUI for local CPC operation. It is connected to a processor hardware LAN, together with SEs from other CPCs that may use this HWLAN. As the next higher systems management level, Hardware Management Consoles (HMCs) can be connected to the processor hardware LAN. Within a hardware LAN, an HMC represents a single point of control for the CPC objects defined to it. HMC users can log in directly at the console, or they can use its Web interfaces to log in. In a hardware LAN environment, multiple HMCs can coexist, either sharing or splitting the control of the CPCs attached to it.

With an HMC, the normal manual CI operation is done by using the Operating System Messages task. One or multiple image objects (LPARs) can be selected, which can be located on different CPC objects. Each selected LPAR allows the use of its integrated console by clicking the desktop message window tab of this LPAR. This allows the operator to view the individual message streams and to send commands to the operating system running in this LPAR. For more information refer to the Hardware Management Operations Guide of your processor.

Manual CI operation of the SE is possible, by either accessing the SE unit located in the CPC cage, or by using the Single Object Operation Task from an HMC to control the SE remotely. These methods however are not considered to be for normal operations. They are used for CPC/SE configuration tasks or for service. For more information refer to the Support Element Operations Guide of your processor.

CI Differences to 3270-Based Console Devices

Compared to 3270 display devices, CI does not provide 3270 data stream related features such as extended color or program function key support. In case of a SE outage, the CI for all CPC LPARs is affected. The CI becomes available again, once an alternate SE is activated as the primary, or the primary SE is reactivated. In a channel-attached 3270 operator console environment, failing consoles can be backed up by using multiple operator consoles over different channel paths.

CI Performance Factors

The CPC's microcode must handle the CI message requests from all its LPARs concurrently. Depending on the number of LPARs and the number of messages that are sent by each operating system over CI, upcoming workload peaks can influence the overall CI performance. This also applies to a SE/HMC, when a varying number of applications have to be serviced, by sending a varying number of CI message events.

On the SE side, CI is lower in priority than time-critical SE tasks such as power and thermal management, and when the SE is busy with those tasks, CI an be slowed down. the activation of an LPAR can affect the CI performance of adjacent LPARs on the same CPC. See also “Testing CI Performance for SNMP Connections” on page 198.

Network Dependencies

CI-based automation with ProcOps depends on the availability of a Internet Protocol network infrastructure. The connection between the SE/HMC and a SA z/OS ProcOps FP system requires this.

If a network element, such as the IP stack on the ProcOps FP system, is not available, CI-based automation cannot work. This also applies if LAN routers or bridges that are used to interconnect the CPC Hardware LAN with the customer Business LAN have configuration or connection problems, or fail.

For CI over BCPii connections, the following dependencies apply:

As long as all participating system images are running on the same CPC, no external network elements are involved. For SA z/OS managed systems, located on different CPCs of a CPC Hardware LAN, at least one HMC is involved as network element for internal routing purposes. The routing HMC and the routing mechanism are transparent to the BCPii protocol. If multiple HMCs in a CPC HW LAN are configured for routing, each of them can potentially be used for that purpose.

IP Stack Considerations

The SA z/OS ProcOps SNMP (TCP/IP) transport requires an IP stack to be active on the ProcOps FP system. The BCPii transport does not have this requirement. SA z/OS ProcOps supports multiple IP stacks on the FP system on a SE/HMC connection level. You can therefore predefine the IP stack to be used for a specific SE/HMC connection with the SA z/OS customization dialog. If you do not define an IP stack name, the system default stack is used.

Adjusting the Receive Buffer size of the ProcOps FP IP stack is an efficient way to prevent CI events from getting lost. See “ProcOps SNMP Sessions” on page 196 for more details about lost events. SA z/OS ProcOps uses the Receive Buffer size value that is specified in the configuration file of the IP stack. With a larger Receive Buffer size, more CI event data can be queued to the ProcOps FP system IP stack before a Receive Buffer full condition occurs and a negative response must be returned to the SE/HMC.

ProcOps SNMP Sessions

When an SNMP (TCP/IP) connection is established to a SE/HMC, ProcOps uses the session parameter: HWMCA_TOLERATE_LOST_EVENTS. This setting makes sure that a session is not terminated by the console application (HWMCA) if the IP stack of the SE/HMC can no longer send events (CI or others) due to a negative send response returned from the ProcOps FP IP stack.

In this case the event is discarded, but the session remains operational. Without this parameter, the session would terminate, the events would be discarded, and the session would have to be restarted. For more information about the session parameters refer to *IBM Z SNMP Application Programming Interfaces*.

OS Message Format Support with ProcOps/BCPii

With SA z/OS, the CI message ID and message text are the only supported parts of an OS message in ProcOps/BCPii. Available CI attributes, like date and time or system names which can prefix a message line, are not supported. They may however be present in the CI window of the HMC.

Similarly, display attributes, such as held message, priority message, prompt indicators or audible alarm indicators, are ignored when the OS message event data is collected by SA z/OS. The unsupported CI message attributes; date, time, system name and unsupported display attributes; held message, priority message, and the audible alarm may be OS-specific. The common CI format of the operating system environments identified by the SA z/OS hardware interfaces apply to: z/OS, z/VM, z/VSE®, z/TPF, Linux on IBM Z, Coupling Facility Control Code (CFCC), and stand-alone utilities such as SADUMP or the Device Support Facility ICKDSF.

Automating Multi-Line z/OS Messages

Care must be taken when automating z/OS multi-line messages, displayed on CI. Internal z/OS message attributes which identify the different parts of a multi-line message are not available with CI; it can be difficult to identify them explicitly.

Parts of a multi-line message are: Header line, one or more Data lines, and End of message line. With the internal message data format of a multi-line message, available over the z/OS subsystem interface (SSI), you can explicitly access these multi-line message parts. ProcOps/BCPii connections to the HMC/SE are always external connections which cannot register to the z/OS SSI. With ProcOps/BCPii CI multi-line messages are only made available as a number of single message lines in the order that they are displayed on CI.

Limiting the Number of z/OS IPL Messages Displayed on CI

As part of the z/OS Load parameter specification, the initialization message suppression indicator (IMSI) can be chosen to control the suppression of messages and system prompts during initialization.

The IMSI character tells the system whether to perform the following actions during system initialization:

- Display most informational messages
- Prompt for system parameters
- Prompt for the name of the master catalog

See the section Loading the System Software in *z/OS MVS System Commands* for a table that shows the possible values for the IMSI character. The values indicate all possible combinations of the actions that are listed.

Whenever possible, it is recommended that you suppress the display of informational messages to reduce the total number of messages at IPL time. If you plan for z/OS IPL automation do not use informational messages as automation action triggers. Choose only messages that cannot be suppressed, in addition to action or decision operator prompts.

Recommended z/OS Console Settings for CI Usage with SA z/OS

Although not a 3270 console device, z/OS supports certain console characteristics for this facility. In the z/OS literature it is referred to as a system console. Because the system console is a special facility, z/OS allows you to activate and to deactivate its usage. This is done with the z/OS console commands V CN(*),ACTIVATE and V CN(*),DEACTIVATE, entered at the HMC or by automation software.

Once activated, z/OS calls this 'the console is in Problem Determination mode'. Operators or automation software can use it to get command responses and unsolicited messages. The amount of unsolicited messages sent to the z/OS system console (CI) can be controlled by setting its z/OS routing codes.

You can specify the AUTOACT group keyword in the CONSOL*xx* member of the PARMLIB. With an AUTOACT group, the ACTIVATE, DEACTIVATE of the system console can be done automatically.

If you have automation routines to issue commands on the CI after IPL is complete, make sure that the allowed routing codes for the system console are limited. Issue command V CN(*),ROUT=NONE on the CI to achieve this. This setting makes sure that you receive only the command responses, job start/stop information, and z/OS priority messages. For more information about system console (CI) and AUTOACT usage refer to *z/OS MVS Planning: Operations*.

Using CI in a z/OS Sysplex Environment

In a sysplex environment you can set the message scope for the system console to cover multiple or all systems of the sysplex.

Do not do this if you use SA z/OS ProcOps or GDPS to monitor and control the systems. The scope must be limited to the system, to which the system console is attached.

The z/OS ROUTE command allows you to forward operator commands from the System Console (CI) of one system in a Sysplex to another system in the same Sysplex for execution. The command response is then returned to the System Console where the ROUTE command was entered. In a SA z/OS environment, do not use the ROUTE command in your CI communication-based automation. Instead, you should establish a connection to the CI of each system and address each target system directly.

The reason for this restriction is the fact that the SA z/OS Hardware interface automatically prefixes CI messages with the processor (dot) LPAR name of the CI, where the message is displayed. For a ROUTE command response, however, this may not be the system location where the response came from.

Running with the z/OS System Console Deactivated

In deactivated mode, the z/OS System Console (CI) does not allow you to issue regular operator commands. Unsolicited z/OS messages are not displayed, with exception of z/OS priority messages. In addition you can:

- Send a message to the System Console from TSO or another z/OS consoles (MCS/SMCS/EMCS), using the system console's z/OS console name as destination,
- Respond to pending system requests (reply numbers). Care must be taken when doing this because no response messages are displayed. In deactivated mode you can also not issue a z/OS D R command to determine the pending requests.

z/OS Health Checker Considerations

The Health Checker MVS component allows monitoring of certain active settings for the System Console (CI) and to issue exception notification messages if they deviate from predefined best practices settings.

Together with many other checks of the system environment the z/OS Health Checker can help to recognize potential system problems or even to prevent system outages.

If you have z/OS system images controlled remotely with the SA z/OS hardware interfaces and you have their System Consoles (CI) running in PD mode, you have to decide if this is really considered to be an

exception in case the IBMCNZ Syscons checking is active. For more information about Health Checking refer to *IBM Health Checker for z/OS: User's Guide*.

CI Security with SA z/OS

You can control the usage of CI with SA z/OS by restricting the user access to the processors hardware and LPARs.

SA z/OS users without the required permission are not able to issue Hardware interface commands either directly with ProcOps or indirectly using a GDPS command which issues hardware interface commands internally. For more information see "Controlling Access to the Processor Hardware Functions" on page 180.

Note: Regardless of restricting the CI access with SA z/OS, some operating systems that use CI as a console facility restrict console usage by requesting an operator to log in first. If you perform such a login with SA z/OS, for example using the ProcOps ISQSEND API command, password information is not protected.

Testing CI Performance for SNMP Connections

Sending a specified number of predefined (pattern) messages to the integrated console using a message per second rate of your choice is the basic logic to determine the overall CI message throughput and performance of a SA z/OS SNMP connection to a SE or HMC.

Once the messages arrive at the ProcOps FP system, they are written to the NetView log. You can determine if OS message events are lost by controlling the message sequence numbers.

In the example shown in Figure 13 on page 198, the ISQ999I message sequence number is 00004. The test case was started for a total of 00010 messages. In the ProcOps FP Netlog you should find all messages from 00001 to 00010. If one or more messages are missing, this indicates that message events were lost on the connection.

```
           1         2         3         4         5         6
----+----0----+----0----+----0----+----0----+----0----+----0----+

+ISQ999I 12:24:01 Test Message 00004 of 00010 *** 1234567890$%&/(


    7         8         9        10        11        12
----0----+----0----+----0----+----0----+----0----+----0

)=? qwertzuiop_QWERTZUIOP* _ProcOps-SYSCONS_ asdfgh+120
```

Figure 13. ISQ999I Test Message Pattern Example

Two REXX program utilities, ISQWTO3 and ISQTSND3 are delivered with the SA z/OS sample library SINGSAMP as members INGEI005 and INGEI006.

Both programs require specifying the total number of messages to be produced on the integrated console (CI) per call. The second parameter can be used to specify the message per second rate that the utility should try to achieve. For installation and usage information refer to the utility source members in the SINGSAMP library.

ISQWTO3 is the utility implementation for NetView environments; ISQTSND3 is a TSO implementation, if a NetView/SA z/OS environment is not available on the z/OS system to be tested.

Run the programs with different combinations of total message numbers and message per second rates. This allows you to emulate different CI message load situations.

Warning!: The usage of these utilities can produce many messages in the system log of the targeted system and the NetView log of the ProcOps FP system.

Summary: Managing CI Performance for SA z/OS

Bear in mind the following recommendations:

1. Follow the recommendations in this chapter to reduce the number of CI messages.
2. If possible, do not use CI alone to monitor the control a system completely. Limit its usage to system initialization and recovery situations.
3. Avoid issuing commands over the CI that may return a large amount of output.
4. For SNMP connections, consider using separate IP stacks with tailored Receive Buffer sizes to cover lost message event situations.
5. Use the ISQWTO3 and ISQTSND3 utilities from the SA z/OS sample library to test peak message load situations and how they affect CI performance.

Appendix B. Migration Information

This appendix provides information about migrating to Z System Automation 4.3 from previous releases. The actions that are required depend on which release you are migrating from.

Migration Steps to SA z/OS 4.4

Before you begin

Before migrating to SA z/OS 4.4, it is recommended that the SA z/OS release you are using runs on the current service level.

Procedure

Complete the following steps to migrate to SA z/OS 4.4:

1. Install the compatibility APAR OA67529 (SA z/OS 4.1 and SA z/OS 4.2 SA z/OS 4.3) before migrating to SA z/OS 4.4, Open the Customization Dialog before converting to an SA z/OS 4.4 policy database in step 2. This APAR also enables you to use a SA z/OS 4.4-built configuration file on a system running SA z/OS 4.1 or SA z/OS 4.2 in a mixed environment.
2. Make a copy of your version 4.x policy database and edit it with the SA z/OS 4.4 Customization Dialog. This converts it to a version 4.4 policy database. For more information, see "Conversion Function" in *IBM Z System Automation Defining Automation Policy*.
3. Read through the following sections before migrating to SA z/OS 4.4:
   - If you are migrating from SA z/OS 4.3, “Migration Notes and Advice when Migrating to SA z/OS 4.4” on page 201
   - If you are migrating from SA z/OS 4.2, “Migration Notes and Advice when Migrating from SA z/OS 4.1” on page 203.
4. Build the configuration files from the policy database with Type=ALL. For more information, see "Building and Distributing Configuration Files" in *IBM Z System Automation Defining Automation Policy*.
5. Load the build files on the designated system. For the first load of the new and converted build files, a NetView recycle is required. For more information, see “Step 17B: Distribute System Operations Configuration Files” on page 109 and the chapter "Building and Distributing Configuration Files" in *IBM Z System Automation Defining Automation Policy*.

Migration Notes and Advice when Migrating to SA z/OS 4.4

This section contains details of various aspects of migration that you must be aware of. Make sure that you read through this section before migrating to SA z/OS 4.4.

Changes to Customization Dialog

This section lists the changes to Customization Dialog of SA z/OS 4.4.

The **ACF Build** option field is introduced in the **Build Options** section of the **Configuration Build** screen. It supports two values such as Legacy or PSP, Default Value is Legacy. The administrator needs to configure, secure, and start the Policy services provider to use the PSP Build Option.

```
  MENU  HELP
------------------------------------------------------------------------------
AOFGBLDP                       Configuration Build
Option ===>

  1 Build a complete enterprise
  2 Build sysplex group or stand alone system
      Sysplex / System name. .                          (*, ?, or name)
  3 Build entry type or entry name
      Entry Type. . . . . . . SYS                       (*, ?, or type)
      Entry Name. . . . . . . *                         (*, ?, or name)
  4 View build report
  5 Delete unused build output members

Build options:
  ACF Build Option. . . . LEGACY      (LEGACY PSP)
  Output Data Set . . . . MULTISYS.SOCNTL
  Mode. . . . . . . . . . ONLINE      (ONLINE BATCH)
  Type. . . . . . . . . . MODIFIED    (MODIFIED ALL)
  Configuration . . . . . NORMAL      (NORMAL ALTERNATE TERTIARY)

Job statement information:  (used for BATCH build)
//AOFRPT   JOB
//*
```

With SA z/OS 4.4, new datasets and ACF Build Option- LEGACY/PSP are added to the INGEBBLD member of the SINGSAMP Sample library. Before submitting this job, you need to edit the job appropriately as described in the sample itself.

```
//*************************************************************
//SYSTSIN  DD *
  ISPSTART CMD(AOFFBBL1 *,+
                        *,+
                        ALL,+
                        *,+
                        1,+
                        0,+
                        DD:PDB,+
                        pdb_entry_name,+
                        pdb_enterprise_name,+
                        DD:SOCNTL,+
                        LEGACY)+
                        NEWAPPL(AOF) BDISPMAX(999999)
/*
```

Post SMP/E Steps

After you perform the SMP/E installation, review the following standard installation steps and, if necessary, carry them out.

Procedure

1. “Step 4A: Update IEAAPFxx” on page 73
2. “Step 4B: Update SCHEDxx” on page 73
3. “Step 4D: Update LPALSTxx” on page 74
4. “Step 4E: Update LNKLSTxx” on page 74
5. “Step 5: Configure SYS1.PROCLIB Members” on page 76
6. “Step 6E: Add the REXX Function Packages to DSIRXPRM” on page 84
7. “Step 9B: Configuring HSAPRMxx” on page 95
8. “Step 12A: Allocate Libraries for the Dialogs” on page 99
9. “Step 16: Compile SA z/OS REXX Procedures” on page 108 (if necessary)
10. “Step 23: Check for Required IPL” on page 117

Miscellaneous

This topic lists the migration notes of some miscellaneous System Automation functions, which you must be aware of.

Changed Messages: AOF700I/AOF702I

To improve the efficiency of service operations, messages AOF700I and AOF702I are updated to better reflect the service level of the CLIST. Additionally, the previously redundant prefix "CLIST" at the beginning of both messages is removed for clarity.

Following syntax is introduced:

AOF700I name release-apar RUNNING ON task WITH PARMS > parmstr

AOF702I name release-apar COMPLETED ON task WITH RETURN CODE rc

- The variable name is the name of the clist.
- The variable release displays the release level of the clist.
- The variable apar displays the apar level of the clist.
- The variable task is the name of the task that the clist ran on.
- The variable parmstr shows the parameters processed.
- The variable rc is the return code given by the clist.

Example for System Automation 4.3 and earlier

AOF700I AOFRAATA V440-OA12345 EXECUTING ON MAC WITH PARMS >

AOF702I AOFRAATA V440-OA12345 COMPLETED ON MAC WITH RETURN CODE 0

Example for System Automation 4.4 and later

AOF700I INGRYST0 V440-NOAPARS EXECUTING ON MAC WITH PARMS >

AOF702I INGRYST0 V440-NOAPARS COMPLETED ON MAC WITH RETURN CODE 0

Migration Notes and Advice when Migrating from SA z/OS 4.1

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 4.2.

Changed Low-Level Qualifiers (LLQs) of Installation Data Sets

The Low-Level Qualifiers of the IBM Z System Automation product have been changed in V4.2.

The following table shows a mapping of the Z System Automation installation data set names between version 4.2 and previous installations.

Table 30. Data Set Names of IBM Z System Automation V4.2 and Earlier Versions

IBM Z System Automation 4.2	**IBM System Automation for z/OS 4.1 or earlier**
Target Library Set (runtime libraries)	
SYS1.SINGDMAP	SYS1.SINGIMAP
SYS1.SINGINST	*SYS1.SINGINST*
SYS1.SINGLINK	SYS1.SINGMOD2
SYS1.SINGLOAD	SYS1.SINGMOD1
SYS1.SINGLPA	SYS1.SINGMOD3

Table 30. Data Set Names of IBM Z System Automation V4.2 and Earlier Versions (continued)	
SYS1.SINGMENU	SYS1.SINGIMSG, SYS1.SINGNMSG
SYS1.SINGMENV	SYS1.SINGMSGV
SYS1.SINGMJPN	SYS1.SINGJMSG
SYS1.SINGOBJV	*SYS1.SINGOBJV*
SYS1.SINGPARM	SYS1.SINGNPRM
SYS1.SINGPDB	SYS1.SINGIPDB
SYS1.SINGPENU	SYS1.SINGIPNL, SYS1.SINGNPNL
SYS1.SINGPJPN	SYS1.SINGJPNL
SYS1.SINGPRF	SYS1.SINGNPRF
SYS1.SINGREXV	*SYS1.SINGREXV*
SYS1.SINGREXX	SYS1.SINGNREX
SYS1.SINGSAMP	*SYS1.SINGSAMP*
SYS1.SINGSENU	SYS1.SINGISKL
SYS1.SINGTENU	SYS1.SINGITBL
SYS1.SINGTREX	SYS1.SINGIREX, SYS1.SINGTREX
SYS1.SINGZFS	SYS1.SINGHFS
Distribution Library Set	
SYS1.AINGDMAP	SYS1.AINGIMAP
SYS1.AINGHFS	SYS1.AINGHFSV
SYS1.AINGINST	*SYS1.AINGINST*
SYS1.AINGMENU	SYS1.AINGIMSG, SYS1.AINGNMSG
SYS1.AINGMENV	SYS1.AINGMSGV
SYS1.AINGMJPN	SYS1.AINGJMSG
SYS1.AINGMOD, SYS1.AINGPGM	SYS1.AINGMOD1
SYS1.AINGOBJV	*SYS1.AINGOBJV*
SYS1.AINGPARM	SYS1.AINGNPRM
SYS1.AINGPDB	SYS1.AINGIPDB
SYS1.AINGPENU	SYS1.AINGIPNL, SYS1.AINGNPNL
SYS1.AINGPJPN	SYS1.AINGJPNL
SYS1.AINGPRF	SYS1.AINGNPRF
SYS1.AINGREXV	*SYS1.AINGREXV*
SYS1.AINGREXX	SYS1.AINGNREX
SYS1.AINGSAMP	*SYS1.AINGSAMP*
SYS1.AINGSENU	SYS1.AINGISKL
SYS1.AINGTENU	SYS1.AINGITBL

Table 30. Data Set Names of IBM Z System Automation V4.2 and Earlier Versions (continued)	
SYS1.AINGTREX	SYS1.AINGIREX, SYS1.AINGTREX

Follow the steps of section "Post SMP/E Steps" on page 202 and update the PARMLIB members, started task JCL, and jobs within your enterprise accordingly.

Changed Commands and Displays

Introduction of new group model

With the introduction of new group model, the following commands and displays are changed:

- INGINFO shows the group model and new optional member policies (MPOS, MPOSOVR, MSEL for all groups regardless of model).
- INGDATA introduces new field for group model.
- INGLIST introduces new **Model** field for group model.
- INGFILT introduces new **APG Model** filter for group model.
- DISPAPG shows the group model in the new **Model** field.
- INGMOVE shows also SYSTEM move groups.
- INGGROUP introduces new **Model** field for group model. MEMBERS panel layout changed to simplify management of availability and satisfactory targets. For relative satisfactory target (*ALL or negative value in the policy), the calculation of the satisfactory target adjustments has been changed to reflect the logic of the automation manager.

Introduction of dynamic resources

With the introduction of dynamic resources, the following commands and displays are changed:

- A new INGDYN command is introduced to provide the capability to view existing templates and their instances, as well as create or delete dynamic instances.
- The INGLIST, INGDATA, and INGFILT commands are extended to support the dynamic attribute of a resource.

Enhanced message management

With the enhancement of message management, the following commands and displays are changed:

- Enhanced code matching options for the ISSUEACT command that now supports the definition of a message severity.
- Improved INGMSGS command that can display an unlimited set of exceptional messages; removed the potential for accidental deletion of write-to-operator-with-reply (WTOR) messages; and provided new command line options to display and manage messages.
- Improved AOFCPMSG command that can avoid capturing messages related to resources that are suspended for automation, and that provides more flexibility when working with messages that need to be deleted (such as DOM).
- Improved INGALERT command that can alert on behalf of critical messages if defined as such in the automation policy. In addition, INGALERT enables the administrator to determine exactly the characteristics and behavior of a clearing event, including the specification to generate no event at all.

Other changed commands and displays

- The INGPAC command introduces a new DETAILS parameter to display the resources that are currently being paced by a single pacing gate, and a new RELEASE parameter to release a specific resource from the command line.
- The INGINFO command introduces a new RSTAT parameter to display the statuses for the resources that are listed as backward and forward relationships.
- The INGCNTL command introduces a new SDF_PARALLEL_UPDATE parameter to turn SDF Focal Point Parallel Update on or off dynamically without policy change and configuration refresh.
- The MDFYSHUT command is enhanced to allow users to control which specific shutdown pass to be processed to speed up shutdown.
- The INGINFO, DISPINFO, DISPMTR, and DISPAPG commands show longer Info Link and Owner fields.

Changes to Command Security

This topic lists the security changes to Z System Automation 4.2 commands and the required migration action.

- The INGSCHED command now supports resource level security. If INGSCHED resource security is enabled, the security administrator needs to grant UPDATE authorization for the resource or schedule before the operator can delete or replace the resource or schedule override. Meanwhile, the security administrator needs to verify that the existing permissions do not use wildcards that cover resource name AND type to avoid that a schedule name matches by accident.
- The INGMDFY command's security profile has changed. If INGMDFY resource security is enabled, the security administrator needs to grant UPDATE or CONTROL authorization for the resource that the operator wants to modify. The caller no longer needs authorization for special resource _CONFIG.
- The INGPAC command's security profile has changed. If INGPAC resource security is enabled, the security administrator needs to grant UPDATE authorization for the pacing gate the operator wants to modify. Existing profiles of pattern `AGT.*.*.RES._MANAGER.PACING` can be deleted.
- The INGVARS command now supports resource level security. If INGVARS resource security is enabled, the security administrator needs to grant UPDATE or CONTROL authorization for the resource that the operator specifies in the INGVARS SET, DEL, or SWAP request.

Changed Exits

This topic lists the exit changes to Z System Automation 4.2.

- The INGGROUP exit AOFEXC13 is changed to provide group model and model 2 specific attributes. For more details, see the sample exit AOFEXC13 in the SINGSAMP library.
- The INGALERT exit AOFEXC17 is enabled to work with the WQE data (for example, jobname) of a triggering message if present. For more details, see the sample exit AOFEXC17 in the SINGSAMP library.
- A new INGAUTO exit AOFEXC27 is introduced that allows to reject the change of agent flags for a resource.
- A new INGDYN exit AOFEXC28 is introduced for programmed actions, such as rejecting creation or deletion of a dynamic resource.
- A new INGDYN exit AOFEXC29 is introduced for programmed actions, such as resuming the new dynamic resource after it's created.

Changes to Customization Dialog

This topic lists the changes to Customization Dialog of SA z/OS 4.2.

- The specification of the entry type Application Group (APG) has been changed. The Define New Entry panel requires specification of **Type**, **Nature**, and **Model** and does not supply any default vales. The values specified in the new processing cannot be changed later. After initial conversion of the PDB to

the new 4.2 level, further changes of the group nature also for existing groups will not be possible. In order to change the nature, you have to delete existing group and create new one with the required **Type**/**Nature**/**Model**.

- With SA z/OS 4.2, some specifications of entry type Application Group (APG) for NEW and UPD have changed. Down-level specifications are not accepted. Use File Update to create a file in the new format and update the NEW and UPD sections accordingly.
- Due to the new layout of CODE definitions for message ID CAPMSGS, the flat file layout also changes. Therefore, to import a flat file created with an older SA version, the layout of CAPMSGS code processing in the flat file must be changed to the following format. Otherwise, syntax errors will be reported during the import.

```
Message ID                          : CAPMSGS
  Description                       :
  Action                            - CODE
    Message ID                      :
    Jobname                         :
    User (CODE3)                    :
    Severity                        :
    Alert                           :
    Inform List                     :
    Comment Type                    :
    Comment Text                    :
    DOM List                        :
```

  After line `Action - CODE` for each 'code matching' record, the bold lines are required.

  Message ID was **Code 1** in the prior format of the flat file. **Jobname** was **Code 2** in the prior format of the flat file. **User (CODE3)** was **Code 3** in the prior format of the flat file. **Severity** was **Value Returned** in the prior format of the flat file.
- Due to the new layout for CODE definitions for message ID CAPMSGS, the PDB must be converted to the new layout. It means that during the initial conversion (first open) of the PDB, the definitions of message ID CAPMSGS are automatically converted to the new layout. This change cannot not be reversed. A PDB already converted cannot be opened with a Customization Dialog of an older SA release.
- Due to the new INGALERT definitions for critical WTORs, an additional record is created in the PDB if a definition with CODE1=CRITICAL_WTOR is available. The new record for message ID INGALERT shows CODE1=WTOR/*, and CODE2, CODE3, and VALUE RETURNED are copied from the existing records.

  It means that during the initial conversion (first open) of the PDB, the definitions of message ID INGALERT are automatically converted to the new layout. This change cannot not be reversed. A PDB already converted cannot be opened with a Customization Dialog of an older SA release.
- If you want to use the generated configuration files on a system with an older SA level, compatibility APARs (at least APAR OA54684) must be installed.

Miscellaneous

Introduction of new application category INGSMFREC

A pseudo application SMF_REC that represents the SMF recording process on the local system has been added to the SA z/OS add-on policies. Therefore, a corresponding category INGSMFREC is introduced to the Application Info policy as an IBM-defined category.

Check whether you use this category in your policy definition and rename it to a value, which is not listed as reserved in the panel help of the Application Info policy.

If you intend to add the SMF_REC application definitions from the *BASE add-on policy, then remove your policy definitions (MVS Z EOD) that stop the SMF recording process.

More precise USS application startup

With SA z/OS 4.2, USS application startup processing is more precise. When ACTIVMSG is called, it will check the real status of the application with category USS before it sets the ACTIVE or UP agent status.

Define a Cleanup Delay independent from the monitor definition

In previous releases, the **Cleanup Delay** defined in the APPLICATION INFO policy item was only honored for those APLs that specified **Monitor Routine** to 'NONE'. The default delay was 12 seconds.

With SA z/OS 4.2, you can define the **Cleanup Delay** whenever needed with or without the **Monitor Routine** defined. The default **Cleanup Delay** is now 0 second. The SHUTFINAL commands are executed immediately after the **Cleanup Delay** expires (no change). Revisit your policy entries of type APL and ADF and consider removing any defined **Cleanup Delay** values to use the new default or explicitly specify a **Cleanup Delay** if you need to delay the execution of the SHUTFINAL commands.

SAP high availability solution

If you use the *SAPSRV add-on policy to automate or manage SAP Remote Application Server resources under USS, complete the following migration steps:

1. Install APAR OA58750.
2. Replace the two scripts `start_as` and `check_as` in the home directory of each `<sapsid>adm` user with the versions of this APAR OA58750. The latest versions of these scripts are included in the `ING_sap.tar` file that is located in the directory `/usr/lpp/ing/SAP`.
3. After the script upgrade, verify the script version by issuing the **`./start_as`** command as `<sapsid>adm` in its home directory. The output should be "`INFO: ./start_as version SAP_v35`". If not, the SAP Remote Application Server resources will stay in STARTED status and finally end up in the DEGRADED status.

Removed predefined leftover NMC messages (OA59461)

The following leftover NMC related status messages have been removed from the System Automation predefined messages.

DUI401I - NMC NETCONV connection has been established (UP status message)

DUI417I - NMC NETCONV connection has been stopped. (Terminated status message)

Coexistence of SA z/OS 4.3 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 4.3. It means that you might be running different releases at the same time.

SA z/OS 4.3 systems can coexist with System Automation 4.2 and 4.1 systems in the same sysplex. Figure 14 on page 209 illustrates this coexistence: it shows a sysplex with three automated systems and a separate automation manager (AM) and its secondary automation manager (SAM).

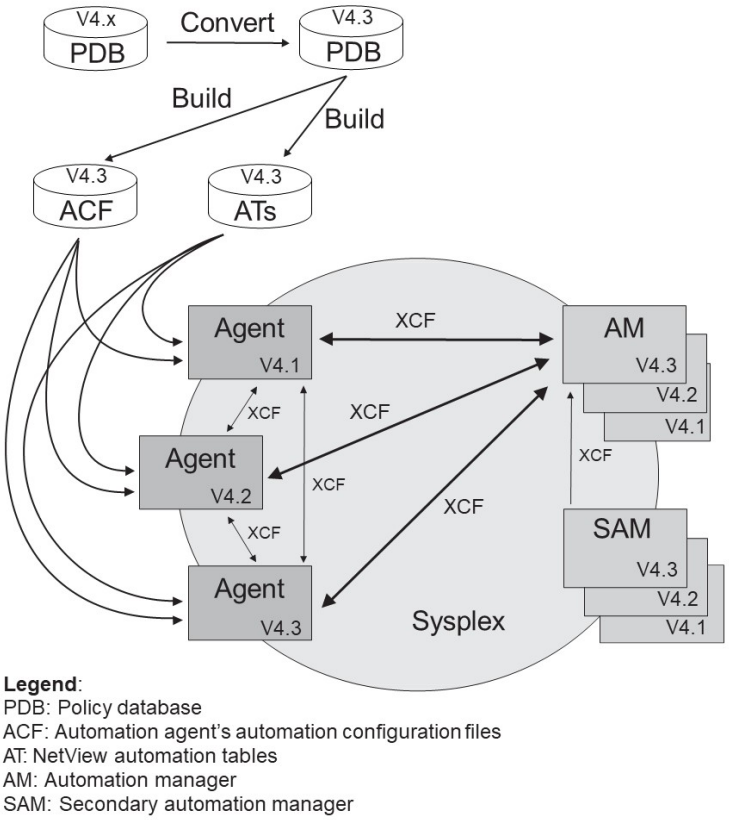


Figure 14. Coexistence of System Automation 4.3, 4.2, and 4.1

Any policy database created by an earlier version of the Customization Dialog (that is, earlier than SA z/OS 4.3) is automatically converted into the SA z/OS 4.3 format when the policy database is opened the first time using the SA z/OS 4.3 Customization Dialog.

The automation configuration files that are built by the SA z/OS 4.3 Customization Dialog can be used by any automation agent running either SA z/OS 4.3, 4.2, or 4.1.

The NetView automation table (AT) that is created by the SA z/OS 4.3 Customization Dialog can be used by automation agents running either SA z/OS 4.3, 4.2, or 4.1.

In a sysplex (that is, the same XCF group), automation agents running SA z/OS 4.3, 4.2, or 4.1 can communicate with an SA z/OS 4.3 automation manager. The communication is via XCF. The automation agents communicate with each other via XCF.

Appendix C. 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.

```
ARMWAIT=nnn
BLOCKOMVS={YES|NO}
BUILDTIMEOUT={ss|180}
CFGDSN=<configuration file data set name>
COMM=XCF
DELAY={ss|0}
DIAGDUPMSG={nnnnn|0}
DIAGINFO=dsname
GRPID={xx|'___'}
IOINTERVAL={n|3}
LEOPT={any}
LIFECYCLE={500|nnnn};MY.AGENT.DATA.SET
LOGSTREAM={YES|NO|GRPID}
NUMQTHDS={n|3}
OVRDELETEDELAY={dd|0}
PREF={n|0}
PROMPT={YES|NO}
SUSPENDFILE=MY.SUSPEND.FILE
START={COLD|HOT|WARM}
STOPDELAY={ss|30}
TAKEOVERFILE=name
TAKEOVERTIMEOUT={nn|12}
FUNCTIONLEVEL={nn|0}
```

ARMWAIT
Maximum number of seconds the automation manager waits for ARM being up during automation manager initialization. Not specified or 0 specified does not cause the AM to wait.

A value from 0-999 seconds may be specified.

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 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.

Notes:

1. 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.
2. For BLOCKOMVS=YES the automation manager must be UID(0).
3. For BLOCKOMVS=YES to work effectively, the stop command for OMVS must be issued as "F OMVS,SHUTDOWN".

NO
If BLOCKOMVS=NO is specified and OMVS shuts down, the automation manager abends due to cancellation by OMVS.

Note:

1. You should not use STOP, DEFER when BLOCKOMVS=NO is specified as it will cause unpredictable results.

BUILDTIMEOUT
May be used to specify a time limit for the completion of the data structure build process that is used during a COLD or WARM start of the primary automation manager. You can specify a value from 0–180 seconds. A value of 180 (3 minutes) is assumed if omitted. A specification of 0 suppresses timing of the data structure build process.

CFGDSN
The CFGDSN value is used only on a COLD start, and may be overridden by an initialization prompt response. On other start types, the default CFGDSN is the one that was in use when automation was last active.

Specify the name of the control data set that contains the SA z/OS configuration that is read by the SA z/OS automation agent and automation manager.

The name can be a fully qualified data set name or a generation data group (GDG) name (either a GDG base name which defaults to generation level 0, or a GDG base name with a level qualifier, for example(-1)).

When you specify a GDG base name and any generation level of the data set is archived, make sure that your storage management product (for example, HSM) is available. If it cannot be guaranteed, specify a generation level which is not archived. Otherwise, you would experience messages HSAM1306I and HSAM1304A.

COMM
This parameter 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 warm start is required when an irrecoverable I/O error occurs while reading from or writing to the takeover file.

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.

FUNCTIONLEVEL
System Automation user function level. You can specify a value up to 5-digit number in the range 0 - 65535. The default value is 0.

This parameter is used to control what capabilities the manager is allowed to perform when operating in a System Automation sysplex environment with systems at different service levels.

Refer to Appendix G, “Function levels,” on page 225 for a detailed description and when to use function levels.

GRPID
Specifies the 2-character suffix that composes the XCF group name that is used by the automation manager and the various agents when communicating among each other.

The value must be the same as specified for GRPID in the corresponding member INGXINIT.

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. At the end of the interval, any deferred I/O is done. The recommended value is 3.

LEOPT
May be used to pass runtime options to the runtime 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 overwritten by installation default settings (CEEOPT) 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. The recommended LE Options.

   The following LE options are recommended for the System Automation Manager:

   ```
   NONIPSTACK(4K,4K,ANYWHERE,KEEP) or THREADSTACK(ON,4K,4K,ANYWHERE,KEEP,512K,128K)
       Note: NONIPSTACK 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 below the line heap need to be tuned.

Notes:

- If an LEOPT=keyword is present in HSAPRM00, it replaces any LEOPT that may have been specified as an input parameter through JCL.
- When specifying options in HSAPRMxx you may have tuned LEOPT statements on multiple 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

The parameter defines if the automation manager establishes a connection to the system logger at initialization time. The default is YES which causes the automation manager to connect to the following log streams:

- HSA.WORKITEM.HISTORY
- HSA.MESSAGE.LOG

You may specify GRPID instead of YES to connect to a different set of log streams:

- HSA.GRP*xx*.WORKITEM.HISTORY
- HSA.GRP*xx*.MESSAGE.LOG

where *xx* represents the value of the keyword GRPID.

Then you may separate the log streams for each subplex. Note that if you specify GRPID but the value of the keyword GRPID is blank, the automation manager returns to the default value YES.

If NO is specified, no access to any SA-related log stream is established and subsequently no data is 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.

Note: Both values, LOGSTREAM and GRPID, must be the same as in the DSIPARM member INGXINIT that is used to start the related NetView agent(s).

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 9 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 secondary automation managers (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 is issued and waits for a response. You can now specify the keyword/value pair:

```
CFGDSN=<fully.qualified.data.set.name>
```

Alternatively you can use a null or 'U' response to indicate that no override values are to be applied.

SUSPENDFILE
This parameter specifies the name of the suspend data set which contains the SA resources that should be suspended by the SA automation manager after loading or refreshing the configuration data set. The name must be a fully qualified data set name. It can also be a member of a partitioned data set.

Per default the SUSPENDFILE parameter is not set and it is a comment within HSAPRM00.

START
Defines the start mode of the automation manager. During initialization, the automation manager retrieves input from:

- 1 The CFGDSN parameter
- 2 Schedule overrides
- 3 The persistent data store (Takeover File)

The following table shows where the automation manager retrieves initialization data for the possible values for the START parameter.

	COLD	WARM	HOT
1	The name of automation manager configuration file is taken from PARMLIB, the START command, or via the PROMPT=YES option.	The last value that was used is taken	The last value that was used is taken
2	Deleted	Taken from the last run	Taken from the last run
3	Deleted	Deleted	Taken from the last run

For a detailed description about what data persisted where and how, see Manager Data persistence during a COLD or WARM start.

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. However, the secondary automation manager reads the CFGDSN parameter from the original HSAPRM*xx* when the SAM was started. Any changes that you make to the HSAPRM*xx* are not reflected in a takeover with a cold start. If you want to perform a cold start with a modified HSAPRM*xx* you must first stop all your SAMs and then restart them.

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. Specify a value in the range 0–999 seconds. The recommended value is 30 seconds.

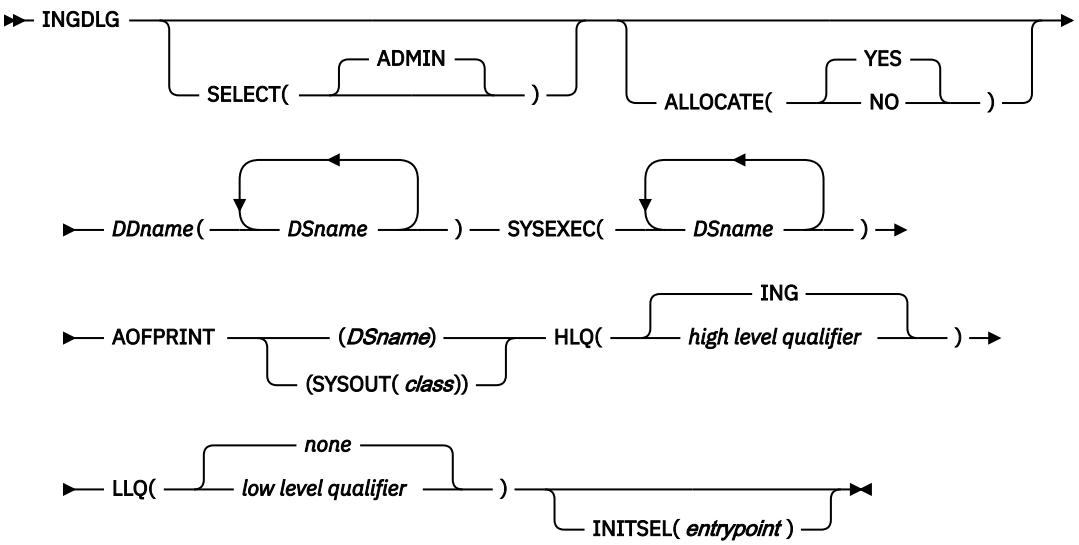
TAKEOVERFILE
This defines the data set name of the takeover file. It must be fully qualified.

TAKEOVERTIMEOUT
The value, *nn*, may range from 1 to 600 seconds. The default is 12 seconds.

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.

Appendix D. INGDLG Command

The INGDLG command allocates required DD names and invokes the ISPF dialog. Its syntax is:



The parameters of the INGDLG command are:

SELECT
If the SELECT keyword is not specified, SELECT (ADMIN) is the default.

ADMIN
Enables the selection of automation policy dialogs. This is the default.

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 the 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 before invoking INGDLG.

***DDname*(*DSname*)**
The *DSname* is the fully-qualified data set name that is to be associated with DD name that is specified. The name is not extended with any prefixes or suffixes that are defined using the HLQ and LLQ parameters.

For example, the following specification allocates the data set ING.CUSTOM.AOFTABL to the DD name AOFTABL:

```
AOFTABL(ING.CUSTOM.AOFTABL)
```

SYSEXEC(*DSname DSname DSname ...*)
For the DD name SYSEXEC multiple data set names are supported:

```
SYSEXEC(DSname DSname DSname ...)
```

This results in the following command:

```
TSO ALLOC ALTLIB ACTIVATE APPLICATION(EXEC)
         DATASET(DSname DSname DSname ...) UNCOND
```

AOFPRINT
For the DD name AOFPRINT, *DSname* is a fully-qualified data set name and the following syntax is 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*.

HLQ
Enables you to change the high level qualifier (HLQ) of the SMP/E data sets, which is currently 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
This parameter can be used to provide a user-selected entry point to the customization dialog. If this keyword is specified, you do not see the Customization Dialog Primary Menu as the first panel when invoking the customization dialog. INITSEL 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 a fast path in the customization dialog, for example:

- To open a PDB in BROWSE mode:

  ```
  INITSEL(BR)
  ```

- To show the Policy Database Selection as initial panel:

  ```
  INITSEL(4)
  ```

- To reach the Entry Name Selection panel for Applications:

  ```
  INITSEL(APL)
  ```

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 deallocation or a failed allocation

Appendix E. Managing IBM Z console availability exceptions

You can use the information in this appendix to set up plans and procedures that help to mitigate the impacts of IBM Z console outages in an Z System Automation environment.

Hardware Management Console characteristics

The Hardware Management Console (HMC) acts as the operational focal point for one or multiple IBM Z mainframes, that are attached to a mainframe cluster. Likewise, System Automation can control multiple IBM Z mainframes with a single connection to an HMC over IP. The HMC gets its CPC and LPAR resources and status information from the Support Elements (SE) of the IBM Z mainframes in the cluster, once the CPC's SE IP addresses are defined or discovered.

In a cluster, you can simultaneously use more than one HMC that have the same or different set of CPCs defined. If one HMC fails, you can use another available one in the cluster to continue operation. This HMC backup scheme can also be configured and used with System Automation. If an SE in the cluster is unavailable, its CPC and LPAR information is not available to all HMCs in the cluster.

You can have your HMCs connected to the mainframe IBM processor LAN, your IP business network, or both. In BCP internal interface (BCPii) transport configurations, as an alternative to the IP protocol, HMCs are used to route BCPii requests and responses between the originator and target CPC Support Elements. This function must be enabled on the HMCs that are supposed to build a redundant BCPii routing pool. Only CPCs in the defined CPC group of the participating HMCs will benefit from BCPii routing. BCPii routing is completely transparent to System Automation and the console SNMP APIs. It is an embedded IBM Z mainframe LAN function.

Support Element characteristics

The Support Element (SE) console acts as the single point of control for one IBM Z mainframe. It is physically connected to the processor and located in a CPC frame. For HMC cluster communication, it is attached to the IBM processor LAN and can be accessed from your IP business network, if configured so.

With Z System Automation, you can control the CPC and its LPARs either through the IP network or by exploiting the BCPii, a direct processor connection support. If the local SE detects that a request does not target the local CPC, the BCPii request is forwarded into the mainframe cluster LAN for HMC routing. In case an SE device fails, it provides its own backup, the stand-by SE.

The SE gets its CPC and LPAR resource and status information directly from the processor hardware and configuration data, which is stored in the activation profiles on the SE hard disk. The outage of an SE affects the control of the attached CPC and all its LPARs. All HMCs in the processor LAN cluster with this CPC defined cannot control this CPC and its LPARs during the SE outage. CPC and LPAR operation itself continue during an SE outage, so operating systems and applications are not affected. However, hardware status changes and events that are emitted by the IBM Z mainframe during the SE outage are lost.

Short-term console outages

There are planned and unplanned outages for SE and HMC consoles. From an SA z/OS perspective, there is no difference between an outage due to a console device error or an access path or network failure that affects the console connection.

For short-term outages, SA z/OS has implemented console polling and monitoring functions, which both can be configured in the SA z/OS processor policy to automatically re-establish a broken or failing console connection.

Standard console connection polling and monitoring

With ProcOps, connection polling can detect consecutive connection restarts that fail and internally prolong the time between the restarts to ten minutes in case a smaller connection polling pace was defined in the SA z/OS PDB.

For SA-BCPii (INTERNAL) connections that are NOT managed by GDPS, the Connection Monitor Interval in the Processor Information policy is used to schedule an SA z/OS internal monitor routine.

For SA-BCPii (INTERNAL) connections that are managed by GDPS, they are exclusively monitored by GDPS. Connection monitoring in SA z/OS is not needed. Therefore, specify the Connection Monitor Interval in the Processor Information policy to NONE to deactivate connection monitoring in SA z/OS.

Planning for longer console outages

Each time the firmware of a console is changed, the console needs restart to activate the change. Such upgrades might be needed in cases of a repair of a previously reported problem, a machine upgrade, or maintaining console firmware serviceability.

If the restart occurs for a console that is defined in your processor policy in SA z/OS PDB, you can take proactive actions to mitigate such predictable outages, which might take several minutes until reboot and console initialization is complete. Some IBM Z maintenance might require repetitive console restarts, resulting in even longer outage periods. Finally, a manual switch from the primary SE to the stand-by SE as part of a console service or recovery action might also cause a longer console outage.

Recommended practices to mitigate predictable outages:

- Know all the users or exploiters, beside operations and administration crews, a console has. It includes knowing which critical systems management components, like SA z/OS, depend on the device. Only then you can judge the real impacts that a coming outage has on your service level agreements or availability targets.
- Ask the IBM customer engineer about the expected outage duration.
- Use an inform policy to make the affected local and remote teams aware of the date, time, and duration of the console outage.
- ProcOps Operations: Depending on your PDB processor definitions and the affected console types, use ISQIPSWT command to switch to an alternate connection, or use ISQXCON command to suspend an unavailable console connection from being used during the outage. If a connection is active again, use ISQXCON to resume it.
- SA-BCPii (INTERNAL) user: Use INGHWSRV command to suspend the currently unavailable console connection from being used. This command can also resume the connection when it is operational again. GDPS has included the INGHWSRV suspend and resume invocation in their user interfaces. GDPS users are advised to use these interfaces to perform suspend and resume operations.

Note: After a console restart, SA z/OS is able to contact the device successfully, but it's possible that not all required CPC and LPAR information that SA z/OS expects is available to the console at this time. It's because the console application itself is still busy collecting that data. Be aware of this additional delay when you plan to resume and restart a console connection. The more CPC or LPAR resources that a console manages, the longer it takes until all resource data is available. SA z/OS can do nothing about this console characteristic.

Consequences of ignoring predictable long console outages

It is highly recommended that you DO NOT IGNORE long console outages and you should consistently suspend affected connections before the console outage. If the connection is suspended, the monitoring stops and the status of the console connection changes to suspended. After the connection is resumed or restarted, automated connection monitoring (if defined in the PDB) is re-established.

If you have not suspended the affected console connection and the connection ends due to a reboot or power-off, the console emits a final event to inform SA z/OS. ProcOps and SA-BCPii sessions are then

terminated. Since the console sessions are not suspended or closed from the SA z/OS side, standard console connection polling keeps retrying to connect the console. As a result, many error messages populate the log files, eventually irritating operators.

However, the strongest impact is that you might jeopardize your systems management obligations, such as disaster recovery commitments or availability targets, by wittingly tolerating 'out-of-control' time periods for the IBM Z resources that are associated with the unavailable console.

Unpredictable console outages overview

At any time, the following incidents might cause longer console outages, preventing SA z/OS services from monitoring and controlling the defined CPCs and LPARs:

- Network problems, including physical connection problems in the customer IP network or IBM's processor LAN.
- IBM Z power problems that affect the attached SE console, when the CPC is operating without a battery backup feature.
- IBM Z battery power problems that affect the attached SE console, when the CPC is operating with battery power.
- Automatic switch to the alternate SE console in a primary SE failure.

Planning automation routines to handle suspend and resume

In a console outage, especially if the connection is in a suspended state, CPC or LPAR hardware operations management cannot be performed, as it will fail immediately. It is your responsibility to add sufficient logic in your routines to avoid this.

SA z/OS has implemented the SUSPEND/RESUME support for IBM Z console connections. You can use the following SA ProcOps commands to suspend and resume the connection path manually or in a user provided automation routine:

- ISQXDST (manual only)
- ISQXCON (manual, automation routine)
- INGHWSRV (manual, automation routine)

Avoiding inconsistent console definitions

You can change object-related data at runtime in the IBM Z consoles and activate such a change immediately. However, it might impact SA z/OS if data values no longer match. The following data must be kept in sync between the console definition and the SA z/OS PDB processor policy that refers to it.

- Processor (CPC) SE console netid and name
- SNMPv3 specific settings, if applicable
- Partition (LPAR) name and (IML) mode
- SNMP API settings: Community names and console IP addresses

Note: Deleting an object on the console immediately affects SA z/OS if the deleted resource is defined in the PDB and a console connection is in use. For example, if you remove a CPC from the defined CPC group of an HMC, while ProcOps is receiving messages and events from LPARs of this CPC, important automation might be broken. A coordinated administration of the console settings and the corresponding definitions in the SA z/OS PDB is the key to avoid such situations.

Avoid outages caused by LPAR security setting changes

Be aware that the IBM Z LPAR security settings can be changed at runtime. The initial partition security settings are done in the activation profile for the LPAR. These settings affect the SA-BCPii console communication, and they do not apply to IP-based communication. The relevant settings are as follows:

- Cross Partition Authority
- BCPii Permissions

System Automation recommends that you have procedures in place to allow a coordinated manual LPAR security setting change on the HMC and possible required changes in the System Automation processor policy. For instance, such a related policy change can be an update in the processor's connection protocol definition or a removal of an LPAR in the processor LPARS and SYSTEMS policy.

In System Automation PDB, the LPAR definitions of CPCs with SA-BCPii connections include the LPAR, where the System Automation instance is running and which issues the BCPii request. The issuing LPAR must have the Cross Partition Authority flag set. If the CPC is a z14 or later, the issuing LPAR can have the BCPii permission set to Send & Receive with the targeted CPC and LPAR in its access list. Alternatively the BCPii permission checking can be disabled to indicate that no permission checking should be made. If the targeted LPAR runs on a CPC that is earlier than z14, BCPii permission settings are not supported and also Cross Partition Authority flag does not apply. If the targeted LPAR runs on a z14 or later CPC and the BCPii permission checking is enabled, at least the BCPii Receive permission must be set, and the CPC-LPAR of the BCPii requester CPC-LPAR must have access set.

Table 31. Issuing BCPii request: Required LPAR settings and characteristics

IBM Z System	OS Type in LPAR	Cross Partition Flag	BCPii Permission Setting
z14 and later	z/OS	Set	Enabled / Send
Earlier than z14	z/OS	Set	N/A

Table 32. Receiving BCPii request: Required LPAR settings and characteristics

IBM Z System	OS Type in LPAR	Cross Partition Flag	BCPii Permission Setting
z14 and later	any	N/A	Enabled / Received
Earlier than z14	any	N/A	N/A

Table 33. Issuing and receiving BCPii request: Required LPAR settings and characteristics

IBM Z system	OS Type in LPAR	Cross Partition Flag	BCPii Permission
z14 and later	z/OS	Set	Enabled / Send & Received
Earlier than z14	z/OS	Set	N/A

If a Cross Partition Authorization flag and/or BCPii permission change at runtime no longer allows a previously started SA-BCPii communication to continue, we have an IBM Z console outage situation. The console APIs do not allow System Automation to predetermine all LPAR security settings, nor does the SE emit an event when an SA-BCPii session can no longer be continued due to changed BCPii permissions.

With this extensive and granular SA-BCPii connection access control, it is important to establish proper change control to prevent SA-BCPii connection outages due to uncoordinated LPAR security setting changes in image activation profiles or at runtime.

Appendix F. Planning to choose feasible CPC names

To identify an IBM Z CPC by name, the console name of the attached SE is used within the console UIs and APIs. This name is factory preset to P*xxxxxxx*, where *xxxxxxx* is the machine serial number. The SE also has a netid, which is factory preset to IBM390PS. The SE console name is always taken as the CPC name in the IBM Z console environment, you cannot change this. However, you can customize the SE console name and the netid. Note, that a Netid.console_name SNA style address will be used internally for BCPii communication although SNA itself is not involved. SA z/OS response reports from the IBM Z hardware also contain this address.

IBM operating systems, IO configuration programs (HCD), and systems management platforms (SA z/OS) tolerate to use CPC names that are not identical to CPC SE console names. IBM recommends that CPC names should not contain mainframe device-type numbers or IBM mainframe brand names, as they could eventually cause additional name change efforts, in case you migrate to a mainframe of different type or brand name. Note that z/OS takes the CPC name of the in-memory IODF data at system IPL time. IODF data sets are created by the IO configuration program.

SA z/OS recommends that you customize the CPC SE console netid in a way that it can be used to identify your own enterprise or internal organization. The SE console name should be shorter than the allowed 8 characters. This SE console name is also used as CPC name in the HCD-IODF configuration utility, so that in the end z/OS uses this name as its local CPC name. SA z/OS and other applications can use the compact name to establish a prefix-appendix scheme to combine the SE console name, which is also the CPC name, with additional criterion markers of your choice, all pointing to the same CPC.

Example

```
        IBM factory set SE console name..P004711
        IBM factory set SE netid.........IBM390PS

      is changed to:
        User customized SE console name..CLOUD7
        User customized SE netid.........MYCORP11

      and used with the IO configuration utility to define:
        HCD CPC name.....................CLOUD7

      which will be used by z/OS as:
        local CPC name...................CLOUD7

      SA z/OS PDB definition:
        PDB processor entry Name.........CLOUD7
        HW Resource Name.................CLOUD7
        Network Name.....................MYCORP11
        ProcOps Target HW Name...........$CLOUD7
```

The previous example illustrates how to use the name CLOUD7 persistently to identify the unique IBM Z mainframe of the MYCORP11 company to the various functions or services that need this name, including SA z/OS. The $ prefix was chosen to enable parallel usage of SNMP and INTERNAL protocols for CLOUD7 in SA z/OS, so ProcOps and for instance GDPS can both monitor this mainframe.

Appendix G. Function levels

To ease migration from one release to another and to reduce the need for compatibility APARs, a function level is introduced that allows administrators to explicitly opt in for using new capabilities.

The functional level is an integer number that represents the capabilities that an automation manager or agent can support. Without any activity on the administrator's side, the capabilities that can be used by System Automation are limited to those that are represented by the current function level.

For the automation manager, its function level is set through the FUNCTIONLEVEL parameter in the PARMLIB member HSAPRMxx (see Appendix C, “Syntax for HSAPRM00,” on page 211). For the automation agent, its function level is set through the FUNCTIONLEVEL parameter in the DSIPARM member INGXINIT (see “Step 6A: Configure NetView DSIPARM Data Set” on page 79).

The following table provides a description of the associated capabilities and deliverables of each function level value.

Table 34. Function level and capabilities

Function level	Capabilities	Deliverable
0	Default content of IBM Z System Automation 4.2.0.	IBM Z System Automation 4.2.0
1	New manager work item that allows more efficient creation of dynamic resources.	IBM Z System Automation 4.2.0 OA59461
2	Default content of IBM Z System Automation 4.3.0 (see Chapter 2, “What's New in 4.4.0,” on page 7).	IBM Z System Automation 4.3.0
3	New INGPLEX SYS and INGPLEX CDS columns. See description of OA63831 for more details.	IBM Z System Automation 4.3.0 OA63831

Note: The function level is a staged initiative and will be rolled out in multiple phases. In this initial phase, the communication between agents and the manager is supported. Before agents attempt to request a function from the manager, they can now first check, whether the manager supports this function and gracefully handle the response.

It is still recommended to use new functions only when all the System Automation releases are on the same level. However, the controlled usage of new functions in such a mixed environment is now facilitated.

Example

Assume that an environment has two systems, S1 and S2. System S1 is on the current release level, while System S2 is on the previous release level. In this example, the capabilities supported by S1 are at function level *n*, whereas the capabilities supported by S2 are on function level *n-1*.

If you want to protect your System Automation from using the capabilities at function level *n*, ensure that both the manager and the agents have an assigned FUNCTIONLEVEL of *n-1*. Once you have upgraded S2 to the current release level, you can safely increment the FUNCTIONLEVEL to *n*.

Appendix H. Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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 jurisdictions 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

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 Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:
© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

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
See automation configuration 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/OSautomation control file, 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.

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.

Advanced Workload Analysis Reporter (zAware)
IBM analytics appliance running in a z Systems® partition, activated in zACI mode. Customers can use the appliance to monitor the console message streams of other LPARs running in the same IBM Z cluster and create trend reports. Exploiting zAware and these trend reports can help to better predict OS outages or performance degradations and initiate proactive clusters.

alert
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.

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 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
See Automation Manager Configuration.

American Standard Code for Information Interchange (ASCII)
A standard code used for information exchange among data processing systems, data communication systems, and associated equipment. ASCII uses a coded character set consisting of 7-bit coded characters (8-bit including parity check). The ASCII set consists of control characters and graphic characters. See also Extended Binary Coded Decimal Interchange Code.

APF
See authorized program facility.

API
See application programming interface.

APPC
See advanced program-to-program communication.

application
In SA z/OS, applications refer to z/OS subsystems, started tasks, or jobs that are automated and monitored by SA z/OS. On SNMP-capable processors, application can be used to refer to a subsystem or process.

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.

application program
A program written for or by a user that applies to the user's work, such as a program that does inventory or payroll.

A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

application programming interface (API)
An interface that allows an application program that is written in a high-level language to use specific data or functions of the operating system or another program.

ApplicationGroup entry
A construct, created with the customization dialogs, used to represent and contain policy for an application group.

ARM
See 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
See American Standard Code for Information Interchange.

ASF
See automation status file.

authorized program facility (APF)
A facility that permits identification of programs that are authorized to use restricted functions.

automated console operations (ACO)
The use of an automated procedure to replace or simplify the action that an operator takes from a console in response to system or network events.

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.

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 the SA z/OS 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 SA z/OS customization dialogs must be used to build the automation configuration file. It consists of:

- The automation manager configuration file (AMC)
- The NetView automation table (AT)
- The NetView message revision table (MRT)
- The MPFLSTxx 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 that SA z/OS is installed on. 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. See also automation configuration file.

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 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.

automation status file (ASF)
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

Base Control Program (BCP)
A program that provides essential services for the MVS and z/OS operating systems. The program includes functions that manage system resources. These functions include input/output, dispatch units of work, and the z/OS UNIX System Services kernel. See also Multiple Virtual Storage and z/OS.

basic mode
A central processor mode that does not use logical partitioning. Contrast with logically partitioned mode.

BCP
See Base Control Program.

BCP Internal Interface
Processor function of IBM Z processor families. It allows for 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, (one or more) timers, and (one or more) 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.

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.

channel-attached
Attached directly by I/O channels to a host processor (for example, a channel-attached device).

Attached to a controlling unit by cables, rather than by telecommunication lines. Contrast with link-attached. Synonymous with local.

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.

CI
See console integration.

CICS/VS
Customer Information Control System for Virtual Storage. See Customer Information Control System.

CLIST
See 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.

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)
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.

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 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.

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 setand XCF couple data set.

coupling facility
The hardware element that provides high-speed caching, list processing, and locking functions in a sysplex.

CP
See central processor.

CPC
See 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.

CPU
Central processing unit. Deprecated term for processor.

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

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 Z System Automation Customizing and Programming*.

D

DataPower® X150z
See IBM Websphere DataPower Integration Appliance X150 for zEnterprise® (DataPower X150z).

DASD
See 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
See double-byte character set.

DCCF
See disabled console communication facility.

DCF
See 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.

device
A piece of equipment. Devices can be workstations, printers, disk drives, tape units, remote systems or communications controllers. You can see information about all devices 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 that allows storage to be directly accessed, such as a disk drive.

disabled console communication facility (DCCF)
A z/OS component that provides limited-function console communication during system recovery situations.

disk operating system (DOS)
An operating system for computer systems that use disks and diskettes for auxiliary storage of programs and data.

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).

display
To present information for viewing, usually on the screen of a workstation or on a hardcopy device.

Deprecated term for panel.

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.

Document Composition Facility (DCF)
An IBM licensed program used to format input to a printer.

domain
An access method and its application programs, communication controllers, connecting lines, modems, and attached workstations.

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 with 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 for NetView customization.

DST
Data Services Task.

E

EBCDIC
See Extended Binary Coded Decimal Interchange Code.

ECB
See event control block.

EMCS
Extended multiple console support. See also multiple console support.

enterprise
The composite of all operational entities, functions, and resources that form the total business concern and that require an information system.

Enterprise Systems Architecture (ESA)
A hardware architecture that reduces the effort required for managing data sets and extends addressability for system, subsystem, and application functions.

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
See Enterprise Systems Architecture.

event
In NetView, a record indicating irregularities of operation in physical elements of a network.

An occurrence of significance to a task; for example, the completion of an asynchronous operation, such as an input/output operation.

Events are part of a trigger condition, such that if all events of a trigger condition have occurred, a startup or shutdown of an application is performed.

event control block (ECB)
A control block used to represent the status of an event.

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 Binary Coded Decimal Interchange Code (EBCDIC)
A coded character set of 256 8-bit characters developed for the representation of textual data. See also American Standard Code for Information Interchange.

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.

focal point system
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.

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.

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.

Status focal point system. In NetView, the system to which STATMON, VTAM and NLDM send status information on network resources.

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.

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 (HMC)
A user interface through which data center personnel configure, control, monitor, and manage IBM Z hardware and software resources. The HMC communicates with each defined central processor complex (CPC) through the Support Element.

Hardware Management Console Application (HWMCA)
A direct-manipulation object-oriented graphical user interface that provides a single point of control and single system image for hardware elements. The HWMCA provides 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 group of systems.

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 provides some level of abstraction from assembler language and independence from a particular type of machine. For the NetView program, the high-level languages are PL/I and C.

HLL
See high-level language.

host (primary processor)
The processor that you enter a command at (also known as the *issuing processor*).

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.

HWMCA
See Hardware Management Console Application.

Hypervisor
A program that allows multiple instances of operating systems or virtual servers to run simultaneously on the same hardware device. A hypervisor can run directly on the hardware, can run within an operating system, or can be imbedded in platform firmware. Examples of hypervisors include PR/SM, z/VM, and PowerVM® Enterprise Edition.

I

IBM Secure Service Container (SSC)
IBM Z partitions, activated to run in SSC operating mode, provide the basic infrastructure runtime and deployment support for firmware or software based appliances, such as zAware or z/VSE VNA.

IBM System z Application Assist Processor (zAAP)
A specialized processor that provides a Java execution environment, which enables Java-based web applications to be integrated with core z/OS business applications and backend database systems.

IBM System z Integrated Information Processor (zIIP)
See Integrated Information Processor (IIP).

IBM Websphere DataPower Integration Appliance X150 for zEnterprise (DataPower X150z)
A purpose-built appliance that simplifies, helps secure, and optimizes XML and Web services processing.

IBM Workload Scheduler (IWS)
See ZWS.

IBM Z Workload Scheduler (ZWS)
The scheduler that plans, executes, and tracks jobs in z/OS environments. It's previously called IBM Workload Scheduler for z/OS (IWS), IBM Tivoli Workload Scheduler for z/OS (TWS), or OPC/A.

I/O resource number
Combination of channel path identifier (CHPID), device number, etc. See internal token.

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
See Information Management System.

IMS/VS
See 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 (IMS)
Any of several system environments available with a database manager and transaction processing that are capable of managing complex databases and terminal networks.

Information Management System/Virtual Storage (IMS/VS)
A database/data communication (DB/DC) system that can manage complex databases and networks. Synonymous with Information Management System.

initial microprogram load
The action of loading microprograms into computer storage.

initial program load (IPL)
The initialization procedure that causes an operating system to commence operation.

The process by which a configuration image is loaded into storage at the beginning of a workday or after a system malfunction.

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 files and that prerequisite applications are functional.

input/output configuration data set (IOCDS)
A configuration definition built by the I/O configuration program (IOCP) and stored on disk files associated with the processor controller.

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.

Integrated Information Processor (IIP)
A specialized processor that provides computing capacity for selected data and transaction processing workloads and for selected network encryption workloads.

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. See also Time Sharing Option.

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
See input/output configuration data set.

IOSP
See input/output support processor..

IPL
See initial program load.

ISPF
See Interactive System Productivity Facility.

ISPF console
You log on to ISPF from this 3270-type console to use the runtime panels for SA z/OS customization panels.

issuing host
The base program that you enter a command for processing with. See primary host.

J

JCL
See job control language.

JES
See job entry subsystem.

JES2
An MVS 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. See also job entry subsystem and JES3

JES3
An MVS 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 complexes that have several loosely coupled processing units, the JES3 program manages processors so that the global processor exercises centralized control over the local processors and distributes jobs to them using a common job queue. See also job entry subsystem and JES2.

job
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.

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)
An IBM licensed program that receives jobs into the system and processes all output data that is produced by jobs. In SA z/OS publications, JES refers to JES2 or JES3, unless otherwise stated. See also JES2 and JES3.

K

Kanji
An ideographic character set used in Japanese. See also double-byte character set.

L

LAN
See 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
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). See synchronous data link control.

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 on z Systems
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)
A network in which a set of devices is connected for communication. They can be connected to a larger network. See also token ring.

A network that connects several devices in a limited area (such as a single building or campus) and that can be connected to a larger network.

logical partition (LP)
A subset of the processor hardware that is defined to support an operating system. See also logically partitioned mode.

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 and system services control point.

logical unit 6.2 (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 peer relationship between session partners
- Efficient use of a session for multiple transactions
- A comprehensive end-to-end error processing
- A generic application program interface (API) consisting of structured verbs that are mapped to a product implementation

Synonym for advanced program-to-program communication.

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
See logical partition.

LPAR
See logically partitioned mode.

LU
See logical unit.

LU 6.2
See 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. See logical unit 6.2.

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.

M

MAT
Deprecated term for NetView automation table.

MCA
See Micro Channel architecture.

MCS
See multiple console support.

member
A specific function (one or more modules or 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 (based on this message list, messages are automated and/or suppressed from z/OS console display), 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
See message processing facility.

MPFLSTxx
The MPFLST member that is built by SA z/OS.

multi-MVS environment
physical processing system that is capable of operating more than one MVS image. See also MVS image.

multiple console support (MCS)
A feature of MVS that permits selective message routing to multiple consoles.

Multiple Virtual Storage (MVS)
An IBM operating system that accesses multiple address spaces in virtual storage. The predecessor of z/OS.

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 systems reside on one or more processors. Or one or more processors can communicate with programs on the other systems.

MVS
See Multiple Virtual Storage.

MVS image
A single occurrence of the MVS operating system that has the ability to process work. See also multi-MVS environment and single-MVS environment.

MVS/ESA
Multiple Virtual Storage/Enterprise Systems Architecture. See z/OS.

MVS/JES2
Multiple Virtual Storage/Job Entry System 2. A z/OS subsystem that receives jobs into the system, converts them to an 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.

N

NAU
See network addressable unit.

See network accessible unit.

NCCF
See Network Communications Control Facility..

NCP
See network control program (general term).

See Network Control Program (an IBM licensed program). Its full name is Advanced Communications Function for the Network Control Program. Synonymous with ACF/NCP.

NCP/token ring interconnection
A function used by ACF/NCP to support token ring-attached SNA devices. NTRI also provides translation from token ring-attached SNA devices (PUs) to switched (dial-up) devices.

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.

NetView (NCCF) console
A 3270-type console for NetView commands and runtime panels for system operations and processor operations.

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 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 that NetView records events relating to NetView and SA z/OS activities in.

NetView message table
See NetView automation table.

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. See logical unit 6.2.

NetView-NetView task (NNT)
The task that a cross-domain NetView operator session runs under. 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.

network
An interconnected group of nodes.

In data processing, a user application network. See SNA network.

network accessible unit (NAU)
In SNA networking, any device on the network that has a network address, including 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.

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 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.

network control program (NCP)
A program that controls the operation of a communication controller.

A program used for requests and responses exchanged between physical units in a network for data flow control.

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.

NIP
See nucleus initialization program.

NNT
See 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.

NTRI
See 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.

OPC/A
See Operations Planning and Control/Advanced.

OPC/ESA
See 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 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/Enterprise Systems Architecture (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
A person who keeps a system running.

A person or program responsible for managing activities controlled by a given piece of software such as z/OS, the NetView program, or IMS.

A person who operates a device.

In a language statement, the lexical entity that indicates the action to be performed on operands.

operator console
A functional unit containing devices that are used for communications between a computer operator and a computer. (T)

A display console used for communication between the operator and the system, used primarily to specify information concerning application programs and to monitor system operation.

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.

optimizer
A special-purpose hardware component or appliance that can perform a limited set of specific functions with optimized performance when compared to a general-purpose processor. Because of its limited set of functions, an optimizer is an integrated part of a processing environment, rather than a stand-alone unit.

OS
See operating system.

OST
See 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
The portion of a panel that is shown on a display surface at one time.

To transfer instructions, data, or both between real storage and external page or auxiliary storage.

panel
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.

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.

In computer graphics, a display image that defines the locations and characteristics of display fields on a display surface. Contrast with screen.

parameter
A variable that is given a constant value for a specified application and that may represent an application, for example.

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.

Data passed to a program or procedure by a user or another program, specifically as an operand in a language statement, as an item in a menu, or as a shared data structure.

partition
A fixed-size division of storage.

In VSE, a division of the virtual address area that is available for program processing.

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, and 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
A processor controller. Also known as the support processor or service processor in some processor families.

PDB
See policy database.

PDS
See 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, utilities, and maintenance boundaries. Contrast with single-image mode.

PLEXID group
PLEXID group or "extended XCF communication group" is a term used in conjunction with a sysplex. The PLEXID group includes System Automation Agents for a subset of a sysplex or for the entire sysplex. It is used to provide XCF communication beyond the SAplex boundaries. For a detailed description, refer to "Defining the Extended XCF Communication Group" in *IBM Z System Automation Planning and Installation*.

POI
See program operator interface.

policy
The automation and monitoring specifications for an SA z/OS enterprise. See *IBM Z System Automation Defining Automation Policy*.

policy database
The automation definitions (automation policy) that the automation administrator specifies using the customization dialog is stored in the policy database. Also known as the PDB. See also automation policy.

POR
See power-on reset.

port
System hardware that the I/O devices are attached to.

An access point (for example, a logical unit) for data entry or exit.

A functional unit of a node that data can enter or leave a data network through.

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.

power-on reset (POR)
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. See initial microprogram load.

PP
See physical partition.

PPI
See program to program interface.

PPT
See primary POI task.

PR/SM
See Processor Resource/Systems Manager.

primary host
The base program that you enter a command for processing at.

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
A device for processing data from programmed instructions. It may be part of another unit.

In a computer, the part that interprets and executes instructions. Two typical components of a processor are a control unit and an arithmetic logic unit.

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 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 logically partitioned mode.

ProcOps
See 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 CICS, Db2, IMS, IBM Workload Scheduler (formerly called *features*).

program operator interface (POI)
A NetView facility for receiving VTAM messages.

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.

PSM
See ProcOps Service Machine.

PU
See physical unit.

R

RACF
See Resource Access Control Facility.

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 workstation from that user can log on to a domain from, that is, to the servers belonging to the domain, and use network resources. Users can access the shared resources and use the processing capability of the servers, thus reducing hardware investment.

resource
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.

In NetView, any hardware or software that provides function to the network.

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 Measurement Facility (RMF)
A feature of z/OS that measures selected areas of system activity and presents the data collected in the format of printed reports, System Management Facility (SMF) records, or display reports.

restart automation
Automation provided by SA z/OS 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 configuration file.

Restructured Extended Executor (REXX)
A general-purpose, high-level, programming language, particularly suitable for EXEC procedures or programs for personal computing, used to write command lists.

return code
A code returned from a program used to influence the issuing of subsequent instructions.

REXX
See Restructured Extended Executor.

REXX procedure
A command list written with the Restructured Extended Executor (REXX), which is an interpretive language.

RMF
See Resource Measurement Facility.

S

SAF
See Security Authorization Facility.

SA IOM
See System Automation for Integrated Operations Management.

SAplex
SAplex or "SA z/OS Subplex" is a term used in conjuction with a sysplex. In fact, a SAplex is a subset of a sysplex. However, it can also be a sysplex. For a detailed description, refer to "Using SA z/OS Subplexes" in *IBM Z System Automation Planning and Installation*.

SA z/OS
See “IBM Z System Automation” on page 257.

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.

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 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 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
See status display facility.

SDLC
See synchronous data link control.

SDSF
See 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.

Security Authorization Facility (SAF)
An MVS interface with which programs can communicate with an external security manager, such as RACF.

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 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 processor (SVP)
The name given to a processor controller on smaller System/370 processors.

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)
A set of protocols for monitoring systems and devices in complex networks. Information about managed devices is defined and stored in a Management Information Base (MIB).

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-MVS environment
An environment that supports one MVS image. See also MVS image.

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
See service language command.

SMP/E
See System Modification Program/Extended.

SNA
See 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
See Simple Network Management Protocol.

solicited message
An SA z/OS message that directly responds to a command. Contrast with unsolicited message.

SSCP
See system services control point.

SSI
See subsystem interface.

start automation
Automation provided by SA z/OS 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 that a subsystem or application is started.

status
The measure of the condition or availability of the resource.

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.

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
See SAplex.

subsystem
A secondary or subordinate system, usually capable of operating independent of, or asynchronously with, a controlling system.

In SA z/OS, an z/OS application or subsystem defined to SA z/OS.

subsystem interface (SSI)
The z/OS interface over which all messages sent to the z/OS console are broadcast.

support element (SE)
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
See service processor.

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
See 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
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.

The full name for SA IOM.

IBM Z System Automation
The full name for SA z/OS.

system console
A console, usually having a keyboard and a display screen, that is used by an operator to control and communicate with a system.

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.

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, 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.

System/390 microprocessor cluster
A configuration that consists of central processor complexes (CPCs) and may have one or more integrated coupling facilities.

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.

T

TAF
See 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
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.

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
A basic unit of work to be accomplished by a computer.

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 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)
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.

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 instruction 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
See 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
See Time Sharing Option.

TSO console
From this 3270-type console you are logged onto TSO or ISPF to use the runtime panels for SA z/OS customization panels.

TSO/E
See Time-Sharing Option/Extended.

TWS
See ZWS.

U

unsolicited message
An SA z/OS message that is not a direct response to a command.

uniform resource identifier (URI)
A uniform resource identifier is a string of characters used to identify a name of a web resource. Such identification enables interaction with representations of the web resource over the internet, using specific protocols.

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.

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 Server
A logical construct that appears to comprise processor, memory, and I/O resources conforming to a particular architecture. A virtual server can support an operating system, associated middleware, and applications. A hypervisor creates and manages virtual servers.

Virtual Server Collection
A set of virtual servers that supports a workload. This set is not necessarily static. The constituents of the collection at any given point are determined by virtual servers involved in supporting the workload at that time.

virtual Server Image
A package containing metadata that describes the system requirements, virtual storage drives, and any goals and constraints for the virtual machine {for example, isolation and availability). The Open

Virtual Machine Format (OVF) is a Distributed Management Task Force (DMTF) standard that describes a packaging format for virtual server images.

Virtual Server Image Capture
The ability to store metadata and disk images of an existing virtual server. The metadata describes the virtual server storage, network needs, goals and constraints. The captured information is stored as a virtual server image that can be referenced and used to create and deploy other similar images.

Virtual Server Image Clone
The ability to create an identical copy (clone) of a virtual server image that can be used to create a new similar virtual server.

Virtual Storage Extended (VSE)
A system that consists of a basic operating system (VSE/Advanced Functions), and any IBM supplied and user-written programs required to meet the data processing needs of a user. VSE and the hardware that it controls form a complete computing system. Its current version is called VSE/ESA.

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 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.

VM/ESA
Virtual Machine/Enterprise Systems Architecture. Its current version is called z/VM.

volume
A direct access storage device (DASD) volume or a tape volume that serves a system in an SA z/OS enterprise.

VSE
See Virtual Storage Extended.

VTAM
See 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.

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
See write-to-operator.

WTOR
See 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
See 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
See extended recovery facility.

Z

z/OS
An IBM mainframe operating system that uses 64-bit real storage. See also Base Control Program.

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.

zAAP
See IBM System z Application Assist Processor (zAAP).

zCPC
The physical collection of main storage, central processors, timers, and channels within a zEnterprise mainframe. See also central processor complex.

Index

Special Characters

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

X

Z

SC34-2716-04