IBM Z System Automation 4.2

Getting Started Guide

SC27-9532-01

Note

Before using this information and the product it supports, read the information in “Notices” on page 123.

Edition notice

This edition applies to IBM Z System Automation (Program Number 5698-SA4) Version 4 Release 2, and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Figures

Tables

About this publication

This manual aims to provide a high-level description of IBM Z System Automation (SA z/OS) V4.2 to help you get started with it. 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.

Intended audience

This publication is intended primarily for the new users of SA z/OS, including automation administrators, operators, and system programmers who install and configure SA z/OS.

Z System Automation Library

Table 1 on page xi 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.
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 Service Management Unite Automation.
Service Management Unite Automation Installation and Configuration Guide	SC27-8747	Describes how to plan, install, set up, configure, and troubleshoot Service Management Unite Automation.

Table 1. Z System Automation library (continued)		
Title	**Form Number**	**Description**
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.

Summary of Changes for SC27-9532-02

This document contains information previously presented in IBM System Automation for z/OS V4.1.0 Get Started Guide, SC27-9532-01.

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

New Information

A new chapter is added to describe the overview, configuration, and usage of the Processor Hardware Interfaces. See Chapter 7, “Processor Hardware Interfaces,” on page 107.

Deleted Information

The Automated System Resource Discovery (autodiscovery) function is dropped in Z System Automation V4.2. So all content about autodiscovery is removed from this book.

Chapter 1. Product introduction-test

This section contains an introduction of SA z/OS and its high-level key concepts.

Product overview

SA z/OS is a NetView-based application designed to provide a single point of control for a full range of system management functions.

SA z/OS plays a key role in supplying high-end automation solutions. It monitors, controls, and automates a large range of system elements spanning both the hardware and software resources of your enterprise.

Monitor

Monitor your resources to respond before they affect end users:

- Monitor hardware components
- Monitor software products and applications
- Monitor automated processes
- Monitor messages and alerts

Control

Take action to control conditions:

- Start and stop your entire enterprise system; initiate hardware and software startup and shutdown sequences
- Manage both remote and local operations and support any IBM Z processor within a Parallel Sysplex
- Manage several operating systems: z/OS, OS/390®, MVS™, VM, VSE, and Linux on System z
- Control a coupling facility as a target system with coupling links in a Parallel Sysplex environment
- React to errors and unscheduled events

Automate

Automate many repetitive and complex tasks:

- Automate failover from one system to another
- Start and shut down software resources
- Start and shut down hardware resources
- Detect and respond to system messages
- Perform initial program load (IPL)
- Perform system power-on reset (POR)
- Build automation policy for your enterprise
- Extend the built-in automation routines by writing your own automation policies

You monitor and control hardware and software resources from, a NetView console, Service Management Unite Automation, or monitor them from Tivoli Netcool OMNIbus.

Product benefits

The following product benefits and services are provided by SA z/OS:

Maximize availability
Leverage high availability and automated operations support for IBM Z and IBM Parallel Sysplex clusters through flexible, cluster-wide, policy-based self-healing. Use SA z/OS to start, stop, and recover applications, systems, and sysplexes. You can also automate applications across multiple sysplexes from a single console.

Minimize outages
Get advanced application disaster recovery capabilities using GDPS to manage remote copy configuration storage subsystems. You can also minimize outages with fast and reliable automatic recovery.

Optimize performance
Optimize system health and performance by applying goal-driven automation to simplify operations, minimize costs, and support business goals. You can reduce automation implementation time and cost by proactively managing availability through performance-driven automation. You can also improve problem resolution by leveraging automated alert notification and escalation.

Automate your enterprise
Monitor and control enterprise-wide processor hardware operations, including powering multiple target processors on and off, resetting them and performing all relevant system management tasks. Respond to messages, monitor status, and detect and resolve wait states, and manage your overall IBM Z processor capacity.

Customize your dashboards
Manage and control applications running on multiple Parallel Sysplexes and Linux with the Service Management Unite (SMU) dashboard from a single point of control. You gain advanced problem resolution with pre-built and in-context commands, and single-step task execution. You can also directly interact with system logs and NetView Canzlog to filter messages and execute commands.

Components

SA z/OS has two key components that are designed to automate system and processor operations. These components are called System Operations (SysOps) and Processor Operations (ProcOps).

System Operations

The System Operations component runs as a NetView application, automates many system console operations and selected operator tasks. These tasks include startup, monitoring, recovery, and shutdown of z/OS resources, components, and applications (including VTAM, RMF, JES2 or JES3, TSO, and others). System Operations can also automate operator console messages, initiate timer-based actions, and prevent critical z/OS resource shortages.

With System Operations, SA z/OS can automate single systems, or automate applications distributed over a sysplex, even across sysplexes or non-IBM-Z platforms, by virtually removing system boundaries for automation through its Automation Manager and Automation Agent design.

Processor Operations

The Processor Operations component monitors and controls processor hardware, LPARs and z/VM guest systems operations.

It provides a connection from a focal point system to a target processor Support Element (SE) or a Hardware Management Console (HMC). 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 activating and deactivating a logical partition, powering on and off, and resetting multiple target processors. You can initiate IPLs and respond to operator prompt messages during system startup, monitor status, and detect and resolve wait states.

Key concepts

This topic introduces the high-level concepts that you need to know while working with SA z/OS.

Automation Manager and Automation Agent

SA z/OS can automate applications distributed over a sysplex, even cross sysplexes or non-IBM-Z platforms, by virtually removing system boundaries for automation through its Automation Manager and Automation Agent design.

Figure 1 on page 3 shows the manager and agent design architecture.

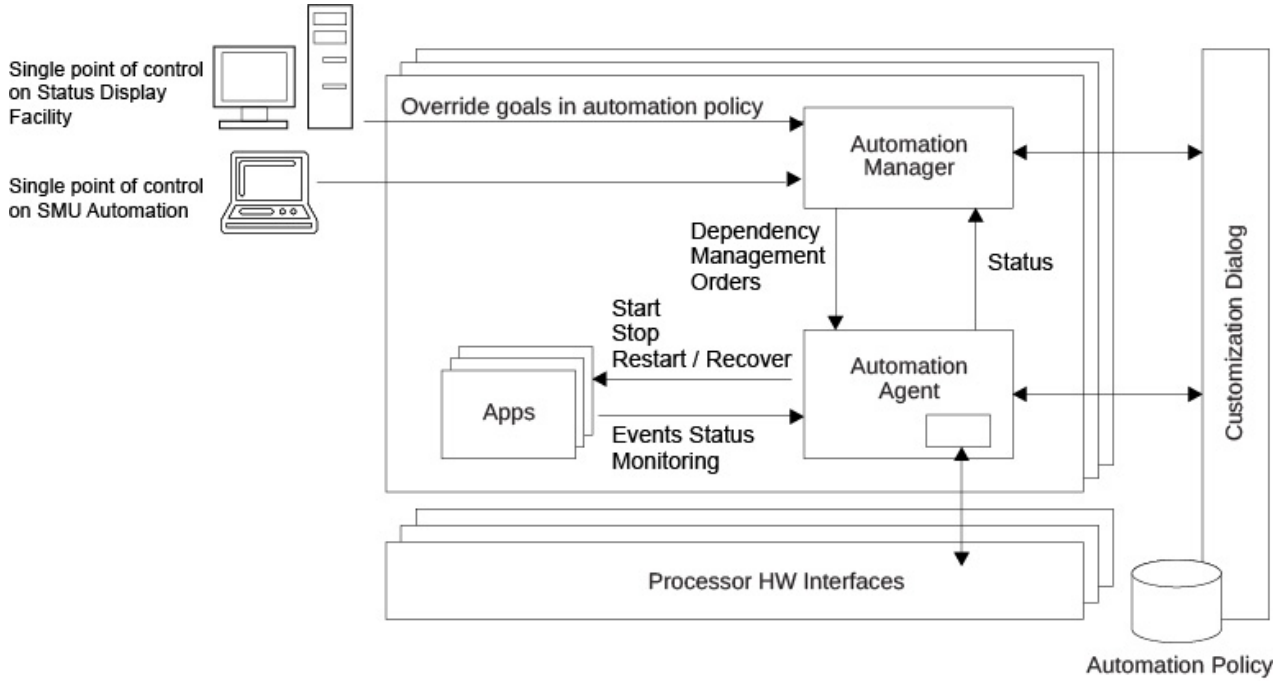

Figure 1. SA z/OS architecture

Automation Manager

The Automation Manager is the central decision maker. Each sysplex has only one active Primary Automation Manager and multiple Secondary Automation Manager for backup. The Primary Automation Manager knows all the applications that SA z/OS controls. Based on policy definitions, it decides when to start or stop applications and sends orders to the Automation Agent, requesting the start or stop of applications.

Automation Agent

Each automated system in the sysplex has an Automation Agent. The Automation Agent receives orders from the Automation Manager and issues commands that are based on the defined automation policy. It runs on the NetView® platform and issues commands to start and stop applications that run on the system. It also monitors the status of applications. Any status changes are sent to the Primary Automation Manager.

Customization Dialog

An Interactive System Productivity Facility (ISPF) application that creates an automation policy and customizes all resources to be automated through System Operations or through the hardware interfaces. The configuration files that are built by the dialog contain the artifacts that are required by both the Automation Manager and the Automation Agent for automated operation purposes. A z/OS enterprise-wide automation policy governs the automated resources and their desired status (goal). The automation

policy for a sysplex or system is loaded under the control of the Automation Manager and used by both, Automation Agent and Automation Manager.

Service Management Unite (SMU) Automation

SMU Automation is a customizable GUI that is available with SA z/OS. It provides a single point of control to operate in your environment. Operators can quickly and confidently analyze, isolate and diagnose problems as all relevant data including important logs is provided in a single place. SMU Automation also enables operators to interact directly with the systems by issuing commands and viewing results without going to a different console.

Policy-based automation

SA z/OS operation is policy-based. Before you can use SA z/OS to automate, monitor, and control the resources in your enterprise, you need to define your enterprise's automation policy.

Automation policy is defined through the Customization Dialog and saved in a policy database (PDB). A single PDB can hold automation policies for all sysplexes in an enterprise. The policy database provides the input to a build process, which creates the applicable automation policies in control files. SA z/OS uses the control files to drive automation within the sysplex. There is one set of control files for each sysplex. All control files for a sysplex must be in the same data set. A stand-alone system is treated as a monoplex.

You can use the Customization Dialog to provide information for SA z/OS, such as:

- Which resources (such as systems or applications) you want to automate, monitor and control
- How resources are to be associated (that is, grouped) with each other for automation
- The dependencies between resources and groups of resources
- What sort of automation, such as automatic startup or shutdown, is to be applied to these resources and how
- The conditions under which automated actions should occur and what actions to perform
- When automation is to be active, and how it handles certain tasks and events during system operation

The power of a policy

- A remarkable characteristic is that SA z/OS does not require user scripts to automate the resources on a system or within a sysplex. Instead it uses a policy that defines the resources, their attributes and the relationships to other resources in the policy. It reduces the need to code rules and scripts that require special skills and are expensive to build and maintain.
- You are able to define automation requirements easily:

  In a policy, you can define which resources belong together and are managed as one (business entity). For example, a Db2 System consists of many resources. With SA z/OS, you can group and aggregate resources to more meaningful manageable entities, for example, My-HumanResource-Application and so forth. You can monitor on this level, issue commands on this level, and manage at the business level rather than at the single IT resource level.
- In policies, you can specify how resources are dependent and related to each other. For example, which of the other resources must be available before a certain resource can be started. In another example, my database must be up and running before my application is started.
- Policy definitions can be reused, copied, and cloned for similar applications elsewhere in the enterprise.
- Because the underlying technology is responsible for the detailed actions, these actions are performed in a consistent and reliable manner. With traditional programming solutions, the testing of abnormal conditions is difficult and prone to be incomplete. The action of automation under these abnormal conditions is, however, critical to the entire automation solution.

Sample add-on policy

SA z/OS comes with a number of sample add-on policies that you can import into an existing policy database to reduce time and effort in creating a policy or updating one. These predefined add-on automation policies are based on best practices, covering many applications, such as IMS, CICS, IBM Workload Scheduler, Db2, SAP, WebSphere, OMEGAMON, and others.

The pictures of these add-on policies can be found in Add-on policies.

Grouping support

Modern applications are often composed of many components, such as data servers, networking, and security components. You can use SA z/OS to monitor and automate resources on the group level.

Grouping of resources can greatly reduce the complexity of automation definitions and operations. If you want to have all members of an application group available, you just tell SA z/OS to have the application group available, rather than each application individually.

An example group is shown in Figure 2 on page 5 , that is, the application group DB21_APG comprises five individual applications.

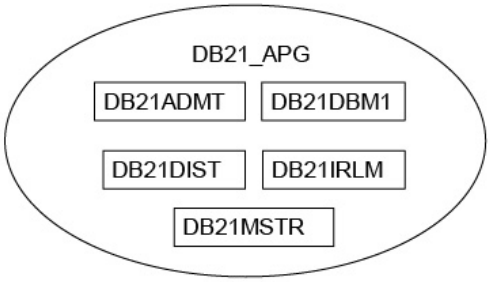

Figure 2. An example application group

A group's status is aggregated from its members' status.

Relationships

Most resources are dependent on services that are provided by some other resources. For example, almost everything uses JES, the VTAM® product, or TCPIP. These dependencies are modeled in your policy as *relationships*.

A relationship runs from the dependent resource to the supporting resources. You can specify a relationship together with a *condition* that must be satisfied for the supporting resource, before the specified action can be processed for the dependent resource. Examples for relationships are HASPARENT, MAKEAVAILABLE (Ma), MAKEUNAVAILABLE. Examples for conditions are WhenAvailable (Wa) or WhenDown.

Figure 3 on page 5 shows that the dependent resource A has a MAKEAVAILABLE/WhenAvailable relationship/condition to the supporting resource B.

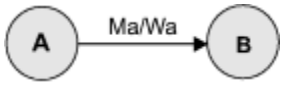

Figure 3. Relationships and Conditions

If a request to become available is issued for resource A, A cannot be made available unless B is available. SA z/OS implements the concept of *request propagation*. A request will be propagated along the dependency graph so that ultimately the original request can be fulfilled. In this example, the request to

A is propagated to B, which can make itself available because it is not dependent on anything else. Then, after B is available, A will become available. The original request is persistent until it is withdrawn.

Figure 4 on page 6 shows another relationship, HASPARENT. Relationships can be expressed on multiple levels. In this example, the CICS application has a dependency on the CICS product, and so does the CICS product to VTAM.

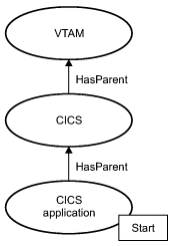

Figure 4. Application Relationships

A HASPARENT relationship embodies two rules:

- The dependent resource cannot be started until the supporting resource is running, that is, the CICS product must be up and running before the CICS application can be started.
- The supporting resource cannot be stopped while the dependent resource is still running, that is, to stop the CICS product, the CICS application must be stopped first.

Through SA z/OS, you can configure relationships in a flexible manner that would otherwise be hard to put in scripted code. For more information, see "Defining Relationships" in *Defining Automation Policy*.

Goal driven automation

The Automation Agent knows how to perform a limited number of actions against each application, usually just Start and Stop.

To complete a complex action, such as recovering from the failure of an application, many actions have to be undertaken in quite a complex sequence. For example, you cannot restart the resources dependent upon the database until the database itself is restarted, which in turn cannot happen until all of the resources that were dependent upon the failed resource are brought down.

Rather than coding these sequences of actions as a fragile script, SA z/OS computes the sequence of actions that are required to perform the recovery based on its goals and its knowledge of the interdependencies between the resources. If something unexpected happens during the recovery, SA z/OS dynamically recomputes its recovery plan and smoothly moves to pursuing it.

SA z/OS manages and recovers through goal driven automation. There are major goals that are set through default desired status values, schedules, and operator requests and minor goals that are set as the desired status on each resource. In response to the major goals, it computes the current desired status for each resource. If the resource is not in its desired state, it simply waits until its dependencies are satisfied and then tells the agent to start or stop it as appropriate. How long does it wait? As long as it takes the dependent resources to be brought into compliance with its goal state. If it is impossible to start the dependent resources, it tells the operator and, if one is available, switches to using an alternate resource or set of resources.

Consider this simple scenario that is illustrated in Figure 5 on page 6:

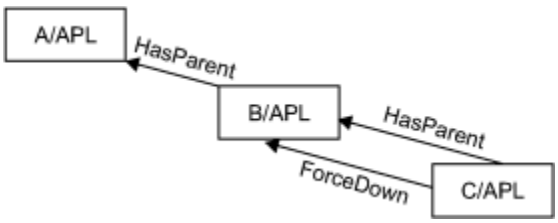

Figure 5. ForceDown Relationships

If resource B fails, SA z/OS first stops resource C, then restarts resource B before it restarts resource C. The only major goal during this scenario is for resource C to be Available, and that is probably an implicit one.

Now consider this scenario that is illustrated in Figure 6 on page 7:

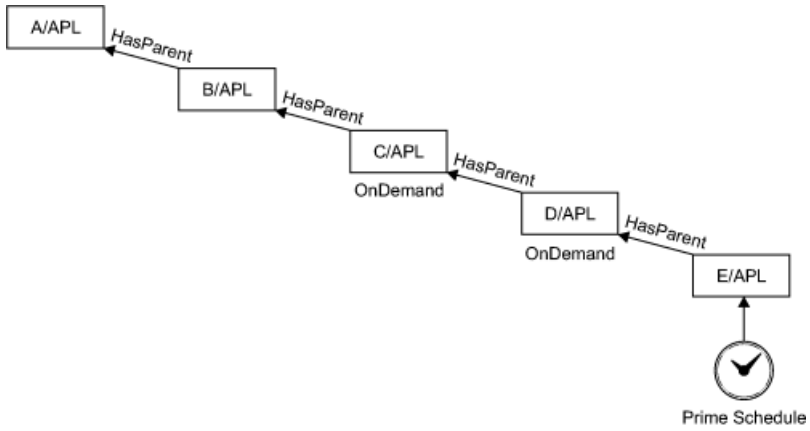

Figure 6. OnDemand Relationships

At 7:00 p.m., the Prime schedule changes from setting an Available goal to setting an Unavailable goal. Then, SA z/OS shuts down resource E, then resource D and finally resource C. Why? Because C and D are defined as only needing to be running when there is a demand for them to be run. With resource E shut down, they are no longer needed. At 11:00 p.m. when the schedule changes again and switches back to setting an Available goal, resources C, D, and E are restarted, in that order.

Chapter 2. Planning

Successful deployment and use of SA z/OS depends on planning and preparedness. Use this information to work through a process of planning for your first SA z/OS deployments.

Roles and responsibilities

This information is written from the point of view of the automation administrator, who is responsible for overseeing the successful installation and deployment of SA z/OS.

The automation administrator is responsible for the listed actions:

- Understand the capabilities and requirements of SA z/OS.
- Negotiate an initial plan for installation and deployment to one or more test systems. Subsequent plans are needed to deploy it to more test and production systems.
- Use the SA z/OS Customization Dialog to model the resources and automation policies on the systems that SA z/OS is going to automate. See Chapter Chapter 5, “Customization,” on page 33
- Activate the SA z/OS programs to load the policy the automation administrator defines and tests.
- Educate the operators about how to best work with SA z/OS.
- Refine, update, and maintain the automation policy for the systems that SA z/OS is deployed to.

During this process (which is expected to take several days), the assistance of a number of other individuals within their organization is required:

System programmers

System programmers have low-level access to the operating system, are able to run SMP/E installations, install procedures and programs and move data. System programmers do all these tasks.

Capacity planners

Capacity planners know how much space and spare compute capacity is available on the customers’ systems and are responsible for the timely provision of more capacity. Capacity planners need to incorporate the deployment of SA z/OS into their plans.

Security administrators

Security administrators define and maintain the security databases on the customers’ systems. Security administrators are needed to create profiles and define permissions so that SA z/OS can be used.

Operators

Operators are the ultimate users of SA z/OS, managing the computer systems that SA z/OS is installed on. Operators are responsible for any manual operations that are required for the computer system, including recovering from any problems or failures that occur. SA z/OS automates their basic duties, so the users need to learn how to work with it to accomplish their goals. For the basic operations tasks, see Chapter Chapter 6, “Operations ,” on page 75

Planning for SA z/OS

SA z/OS has two major components: System Operations (SysOps), and Processor Operations (ProcOps). While there is some commonality in the planning for these components, there are also some steps that are unique to each component.

For the purposes of the chapter, it is assumed that you are installing the SysOps. If you want to deploy ProcOps, review the *IBM Z System Automation Planning and Installation*.

Deploying SA z/OS

Identify initial target systems

About this task

Who: Automation administrator, system programmer, capacity planner

Procedure

1. Identify the system where the system programmer runs the SMP/E installation of SA z/OS.

   For more information, see the Program Directory.
2. Identify the system where the automation administrator runs the Customization Dialog.

   The system must have TSO access, and access to several cylinders of disk storage. The amount of storage depends on the size of the policy. For an installation with 3 systems, 20 cylinders at minimum are preferable. Running the Customization Dialog does not influence the operation of the system it runs on. A reasonable system availability is recommended so that the Customization Dialog can be used as required.
3. Identify the system or systems that SA z/OS automates first.

   The system must be a test system, as it is shut down (intentionally or otherwise) once or twice during testing and training. The system is also not the system where you run the Customization Dialog, as you may need to run the dialog to update the policy to prevent SA z/OS from shutting down the policy. About capacity, SA z/OS requires two copies of the SMP/E target libraries, space for several automation policies, and space for logs and VSAM files. See both the NetView and the *IBM Z System Automation Planning and Installation* for more details.

Establishing naming conventions

Who: Automation administrator, system programmer

As a part of the deployment process, you need to establish a number of naming rules that are related to the installation of SA z/OS:

- High-level qualifier (HLQ) for SMP/E libraries on SMP/E installation system
- High-level qualifier for SMP/E libraries on deployed systems
- High-level qualifier for locally allocated data sets
- High-level qualifier for automation policy databases (PDBs)
- High-level qualifier for SysOps Control (SOCNTL) staging data sets
- High-level qualifier for active automation policy. For your first deployment, use a normal PDSE, but consider the use of GDGs for later deployments.
- Domain name that is used for NetView component (five characters, unique within the SNA network)
- Manager/Agent communication group suffix for unique identification (two characters, unique within the sysplex) of the systems that form an SA-sysplex, also referred to as an SAplex.

Installing and deploying SMP/E

About this task

Who: System programmer, security administrator

Procedure

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 use a system where the Customization Dialog is run and on the systems where SA z/OS is deployed for automation.
3. The system programmer transmits 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 create copies of these data sets as you should never update the data sets that are being used by the product through maintenance.
4. On the system where the Customization Dialog is run, the system programmer makes 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.

   See the *IBM Z System Automation Planning and Installation*
5. The security administrator provides 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 (PDBs) and the System Operations Control File (SOCNTL) staging data sets.

Configuring the target systems

About this task

Who: System programmer, automation administrator, security administrator

Procedure

1. Before SA z/OS is started on a target system, you perform an installation process and configure the system to work with it.
2. The security administrator gives the system programmer and the automation administrator ALTER access to the HLQ for the locally allocated and active automation policy data sets.

   He also needs to give the following access to the user ID that the SA z/OS started tasks are associated with (NetView, System Automation Manager):
   - READ for SMP/E and active automation policy
   - UPDATE for the locally allocated data sets
3. It is recommended that the automation administrator runs the SA z/OS Configuration Assistant tool to configure the product. For manual configuration, refer to *IBM Z System Automation Planning and Installation*.

   A set of tailored jobs, procedures, and parameter files is produced. Then, the automation administrator runs the produced jobs. The system programmer copies the procedures and the parameters file into the system production libraries, and tailors them to or consolidates them with existing procedures, as required. For more information, see “Installing SA z/OS” on page 15
4. The security administrator ensures that the procedures and data sets are properly authorized.

   See Chapter 4, “Security,” on page 31.

Building a minimal automation policy

About this task

Who: Automation administrator, system programmer

Procedure

1. The automation administrator gathers data from the target systems.
2. The automation administrator uses the Customization Dialog to create an Automation Policy Database containing a minimal automation policy for each target system.

   The Policy Database contains details of the address spaces that run there along with the specific policy that is used for that automation. Most automation options in this policy are disabled. The purpose is to check that the installation is successful.
3. When it is complete, the automation administrator builds the SOCNTL data set for the automation model.

Starting SA z/OS and verifying your installation

About this task

Who: Automation administrator, system programmer, security administrator

Procedure

1. Deploy the minimal automation policy (SOCNTL data set) that you built following the instructions in "Installing SA z/OS" on page 15.
2. Continue to follow the instructions to start the SA z/OS Automation Manager and Automation Agent.

   Both the Manager and the Agent come up. No applications are stopped or started.
3. Log on to NetView component.
4. Perform the configuration validation step with the operational command INGAMS described in "Verification" on page 28.

What to do next

It is advisable to have the system programmer and security administrator available to help deal with any problems that are encountered.

Improving your automation policy

Who: Automation administrator, system programmer, operators

The automation administrator now needs to produce a more comprehensive automation policy for the target systems. The automation administrator can either build upon the minimal policy he already has or he can start over using the supplied samples. The system programmer or operator, or both are able to assist to determine data about the system and to decide upon the best policy option to use.

For more information, see Chapter 5, "Customization," on page 33.

Restarting SA z/OS

About this task

Who: Automation administrator

Procedure

1. Copy the SOCNTL data sets over to the target system after your improved automation policy is built.
2. Place them where SA z/OS can find them.
3. Start SA z/OS if it is not already running.
4. Use the INGAMS command to load your improved policy.

What to do next

You can now use the commands that are detailed in Chapter 6, “Operations ,” on page 75 to explore your policy. When there are errors in your policy, you might find that SA z/OS is trying to start or stop items unnecessarily.

Testing your automation policy

Who: Automation administrator

You are now ready to test the Stop and Start command instructions, and dependency definitions for the resources that you define to SA z/OS. There is more work that is involved on the first systems you deploy to and many of the policies are new. As you add more systems into the policy, the testing load is reduced because you can reuse tried and tested policies from earlier systems.

Training your operators

Who: Automation administrator, operator

At a minimum, the operators need to read Chapter 6, “Operations ,” on page 75 and work through it on your test systems. You can also run some training or discussion sessions, or both, to establish local procedures for using SA z/OS. After your operators are trained, they can help you with the testing, which improves their familiarity with the product.

Expanding and maintaining your automation policy

Who: Automation administrator

You can now expand your automation policy, activating or defining automation for more address spaces and testing it as you go. You also need to update the policy if the system programmers add, remove, or change the address spaces that run on the system.

Additional deployment plans

After you are comfortable working with SA z/OS on your test system, you can plan its deployment to further test systems and production systems.

The deployment on other systems is similar. Building on top of the automation policy that you have created for the initial test system, you can incrementally add other sysplexes, systems, and resources to it as necessary. From the policy, you can build an enterprise-wide automation configuration (recommended) or sysplex-wide automation configuration that you load, test, and possibly refine on each environment, step by step.

Planning worksheet

The `INGDOPT Configuration Options` file contains a list of options to configure SA z/OS. Print a copy and you can write in the values agreed upon in your planning meeting.

The comments in the file provide guidance as to the meaning and usage of the values. It is helpful to read “Configuring SA z/OS” on page 15 , which describes the use of the file with the Configuration Assistant and Chapter 4, “Security,” on page 31, which describes the security model.

The output of the Configuration Assistant is normally written to staging data sets from which the system programmer copies/merges/adapts it into the appropriate PROCLIB and PARMLIB data sets. Writing the output directly into the production libraries is not recommended as it could overwrite the output from earlier runs if you are using libraries that are shared between multiple systems.

Chapter 3. Installation and Configuration

This section provides instructions for SA z/OS installation and configuration procedures.

Installing SA z/OS

SA z/OS is installed through System Modification Program Extended (SMP/E) according to the Program Directory. Refer to the Program Directory (GI13-4184) for detailed instructions about the installation procedure.

Configuring SA z/OS

The configuration of SA z/OS is supported by the Configuration Assistant. By using the Configuration Assistant, all NetView-related configuration aspects for SA z/OS are handled automatically for you.

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:

- A tailored readme file $INGREAD containing detailed instructions of which configuration steps must be completed and how to achieve it
- 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 have to 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

Preparing to Configure SA z/OS

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 16.

2. Create a work copy of the INGDOPT Configuration Options file and the Configuration Assistant sample job (INGDCONF). See “Create Work Copies” on page 16.
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 17.
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 17.
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.

Perform this step on the target system, where SA z/OS is to be configured. See the configuration step "Configure the System Logger" 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 need to update the `SMFPRMxx` member.

Perform this step on the target system, where SA z/OS is to be configured. See the configuration step "Update SMFPRMxx" in *IBM Z System Automation Planning and Installation*.

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 need to configure the TSO REXX Function Package on the target system where SA z/OS is to be installed.

See the configuration step "Configure Function Packages for TSO" in *IBM Z System Automation Planning and Installation*.

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*.

IBM Service Management Unite Automation (optional)

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

SMU Automation communicates with z/OS systems that are managed by Z System Automation, through an adapter, which is commonly called 'E2E automation adapter'.

Looking for the following information?	Find the details here
How to install SMU Automation with Docker	• Installing SMU Automation with Docker
How to connect SMU Automation to End-to-End Automation Adapter	• Quick Startup of End-to-End Automation Adapter • Configuring the SMU Automation host
SMU Automation usage, operation tasks, and how to customize dashboards, etc.	• Service Management Unite Automation embedded online help

End-to-End Automation (optional)

SA z/OS provides cross sysplex and cross platform automation capabilities. It allows automating resources spread across different SAplexes or across different platforms similar to resources that are located in a single SAplex or platform. 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 2. 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			

Table 2. Worksheet for job names (continued)

Application	Description	Default Job Name	Real Job Name	Default Procedure Name	Real Procedure Name
RESOLVER	TCP/IP Name Resolver	RESOLVER			
RMF	Resource Measurement Facility	RMF			
RMFGAT	RMF Monitor III Data Gatherer	RMFGAT			
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 4.2 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 4.2 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 4.2 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 4, “Security,” on page 31.
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 = 10/24/19
 Operator ID = JMH             Sysplex = SYSPLEX1          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      SYSPLEX1 INGXSG    V4R2M0  XCF
     SYS1     SYS1$$$$1 PAM   READY      SYSPLEX1 INGXSG    V4R2M0  XCF
```

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

Chapter 4. Security

After the initial configuration, you need to make SA z/OS secure for your production environment. 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.

You are advised to first change the default passwords of the operator IDs that come with the product, following the steps as follows:

1. Locate the default operators that are defined in <nv_hlq_smpe>.DSIPARM member DSIOPFEX.
2. Copy this member to <sa_hlq_user>.DSIPARM
3. Change the PASSWORD parameter for each operator. 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 an SAF product.
- Command authorization is done by an SAF product that is based on the issuer of a command.
- Resource authorization is done by an SAF product that is based on the issuer of particular commands

For additional information about how to fully secure SA z/OS, refer to the "Security and Authorization" chapter in *IBM Z System Automation Planning and Installation*.

Chapter 5. Customization

SA z/OS starts and stops your applications, recovers them if they fail, and coordinates actions across large groups of applications.

To do this, SA z/OS needs a model of your application workload and policy definitions. This information is entered through an ISPF application that is called the Customization Dialog, which is provided as a part of SA z/OS.

The data in the model is stored in a data set known as a policy database (PDB). The PDB cannot be used by the run time environment directly. A build step is required to compile the PDB into another data set called the System Operations Configuration File (SOCNTL), which is used by the Automation Agent and the Automation Manager. The SOCNTL data set is used at run time. Copy the SOCNTL data set to each system that loads the automation policy. The PDB data set, however, remains on the single system where you run the Customization Dialog.

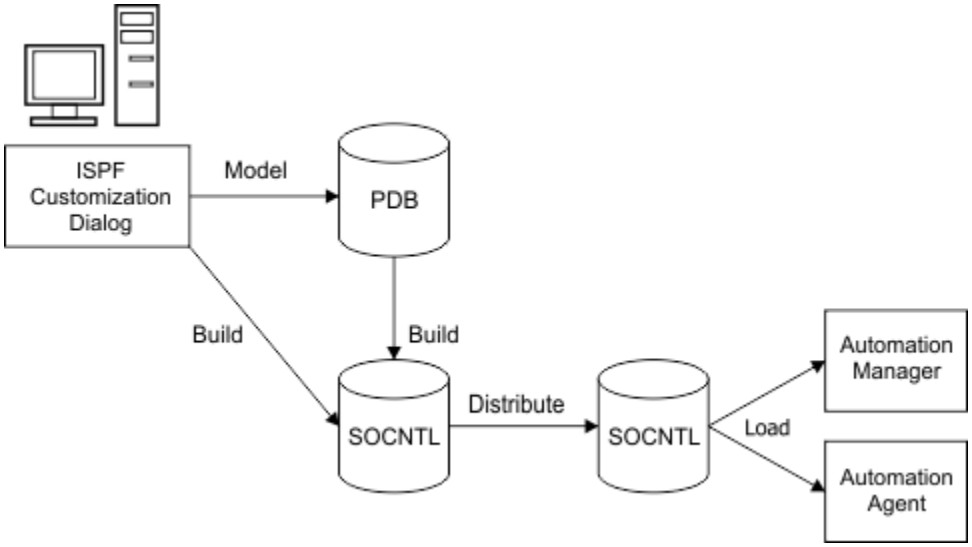

Figure 7. Automation Policy Lifecycle

Customization tools

While the primary modeling tool of SA z/OS is the Customization Dialog, SA z/OS provides a number of extra tools. The tools assist you in gathering data about your systems and creating an automation model that is built around tried and trusted practices.

Customization Dialog

The Customization Dialog is an ISPF application that performs a number of actions upon your policy databases (PDBs). You also edit the data within the PDBs.

The following actions are available to you when working with a policy database. With the listed actions, you can construct and fine-tune an automation model for your systems.

- Create a new empty database or one from templates (add-on policies).
- Create and edit the individual entries within your database.
- Build the configuration files from the data in your database.
- Perform updates with a text or flat file.
- Import data from other databases or from add-on policies.
- Generate a report of your database.

Add-on policies

SA z/OS provides multiple add-on or sample policies that you can use as a basis to start your automation model. You have some options to create your automation model:

- Start with a populated add-on policy and review it to change the definitions to match your own systems.
- Start with a blank model and create your entire model through the editor.

While both approaches have their advantages, it is recommended that you use an add-on policy for your first automation model.

What is in a policy database?

A policy database (PDB) contains the data that models the systems and applications that you want SA z/OS to automate. It also includes the required policy to tell SA z/OS how you want it to automate those resources.

The Customization Dialog supports the definition of the models for multiple systems within the same policy database. The result is a considerable benefit in terms of sharing resource definitions and standardizing the automation of resources across multiple systems.

Figure 8 on page 34 shows a single policy database that is used to define your automation model and then used to produce a System Operations Control File (SOCNTL) data set from that policy database. The SOCNTL data set is then distributed to multiple different systems, where the automation extracts just the resources that it needs to run on that system.

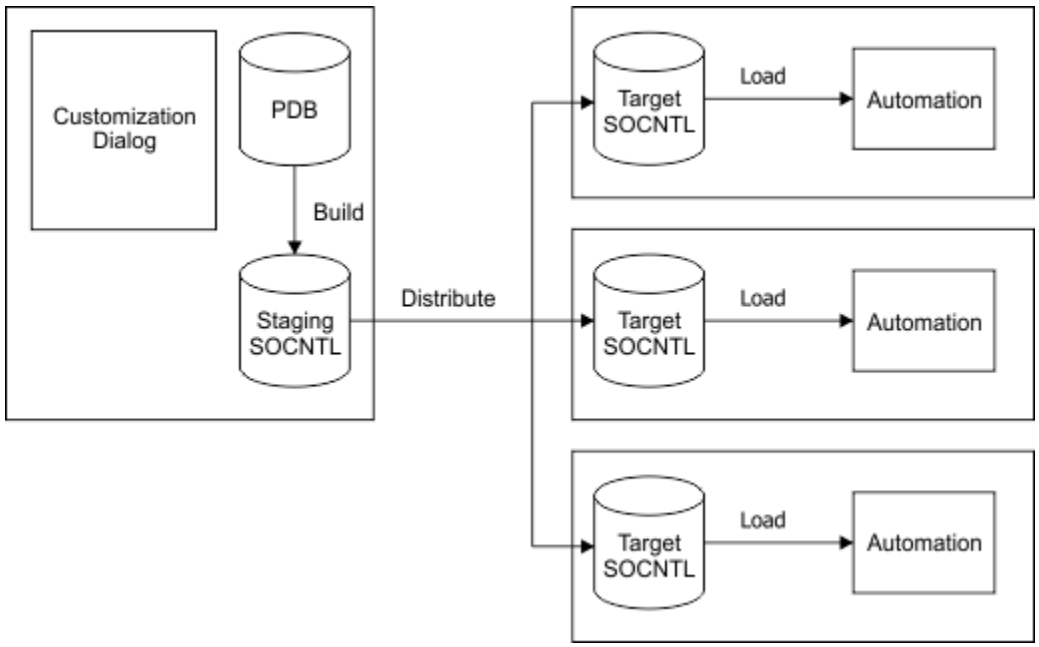

Figure 8. Policy Database and SOCNTL Distribution

You can, for example, define a single set of TSO automation and link the set to every system definition that you deployed SA z/OS to. These systems all run TSO in the same way with the same policies.
If you need to create some systematic variations in naming, SA z/OS provides a symbolic substitution mechanism, called System Automation Symbols (AOCCLONE), to tailor the automation for each system when built by the Customization Dialog policies.

Building blocks

There are many different entity types within a policy database, but you do not have to understand them all. The defaults of most entity types are a good starting point and there are a few that are not required for SA z/OS. The following resources are considered the most important.

Resources

Resource is a generic term that means any actual or logical entity that is modeled by SA z/OS. Address spaces are modeled as APL (Application) type resources, active and passive Monitors are modeled as MTR (Monitor) type resources. Systems are modeled as SYS (System) type resources, logical groupings of applications are managed by APG (Application group) type resources. The Customization Dialog recognizes the abbreviations that are given here for the resource type specified.

General naming conventions are used for all resources:

```
name/type/system
```

For sysplex-wide APGs, there is no `/system` qualifier, so the naming schema is:

```
name/APG
```

Application (APL)

The Application resources represent the individual address spaces (and other entities) that are to be automated.

For example, you might define APL resources for JES, VTAM, and TSO, as shown in Figure 9 on page 36.

Application Group (APG)

An Application Group is a group of Applications. It is used to manage the applications as a whole. For example, as shown in Figure 9 on page 36, you might define an Application Group that is called BASE_SYS, and make the JES, VTAM, and TSO resources members of it.

There are two types of application groups:

- System Application Group, which defines a set of applications and resources that reside on single systems. You need to connect the System APG to each system where it resides respectively.
- Sysplex Application Group, which defines a set of applications and resources that reside on multiple systems within a sysplex. You need to connect the Sysplex APG to the Sysplex Group (GRP), then you can monitor the APG at the sysplex level, rather than at system level.

System (SYS)

A System (SYS) defines a single system that is to be automated. It includes definitions to automatically respond to events that happen on the system.

For example, you might define a System that is called SYS1 as shown in Figure 9 on page 36. Connect the BASE_SYS APG to SYS1 to instruct automation to run the resources in the BASE_SYS APG on the SYS1 System.

Group (GRP)

A Group (GRP) is a collection of one or more systems (SYS). Every collection of systems, whether it consists of only one system or multiple systems, must be represented by a Sysplex Group. An SA z/OS enterprise may have several Groups.

For example, you might define a Group that is called MONOPLEX1 to hold the SYS1 System, as shown in Figure 9 on page 36.

Enterprise (ENT)

The Enterprise (ENT) is the logical top of the tree and contains some global data about your business and global policies that are used by SA z/OS. Each policy database contains only a single enterprise.

If you follow the examples, you now define a relationship as follows:

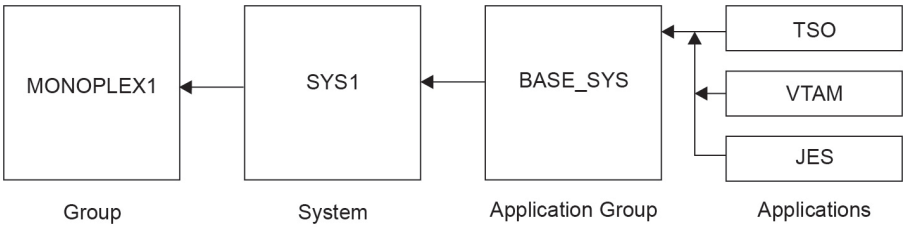

Figure 9. Simple Entry Structure

Building up a structure complex enough to model an actual system is a matter of creating extra Applications and Application Groups. Then, link them into the structure. To extend the automation to cover more systems, you create new System resources and link them into the structure:

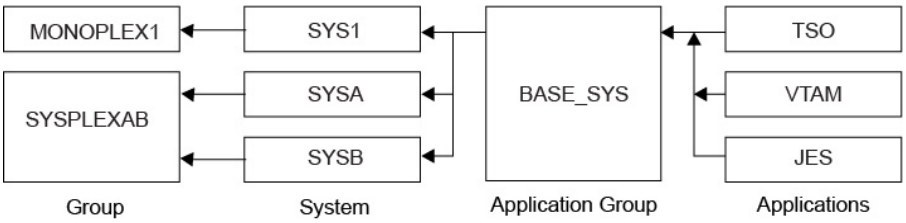

Figure 10. Complex Entry Structure

Cloning definitions

Reusing the definitions between different systems is good practice, but sometimes the definitions are not the same on each system.

The Customization Dialog provides a mechanism to help, using the system automation symbols, AOCCLONE*x* variables (where *x* is 0-9, A-Z). You can assign up to 36 system automation symbols to each system. When you use a system automation symbol in a job name, each system can run with a unique job name while the application or subsystem shares a single automation policy across systems.

For example, you might set the clone value AOCCLONE1 to S1 on SYS1 and S2 on SYS2. You could then modify your VTAM APL so the job name is NET&AOCCLONE1. The result of this is that it would run with the job name of NETS1 on SYS1 and NETS2 on SYS2.

You can provide a description of the clone values under the definition of your Enterprise.

For more information, see “Cloning with symbolic values” on page 55.

Starting the Customization Dialog

Your system programmers install the dialog on the system where you are going to perform your customization activities. The Configuration Assistant provides you with a REXX script called INGEDLG. That script links into your regular ISPF panels, or you issue a command to start them.

After the INGEDLG script starts, you see a panel as follows:

```
  MENU  OPTIONS  HELP
-------------------------------------------------------------------------------
           IBM Z System Automation 4.2 Customization Dialog
Option ===> ____________________________________________________________
                                                                    More:    +
  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
```

Customization Dialog tutorial

The following short tutorial guides you through the basics of creating an automation policy with the Customization Dialog. This tutorial is highly recommended for new users.

Figure 11 on page 37 shows the resources that you're going to define.

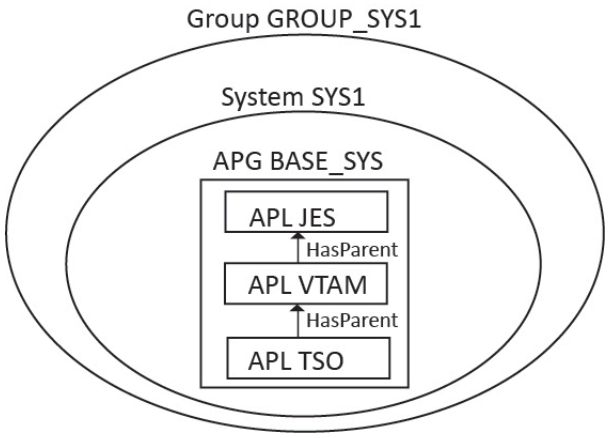

Figure 11. Resources to be defined in the tutorial

Getting help

You can press PF1 for help with the panel that you are on. Pressing it a second time gets you general help with the dialog. To exit the help panel, press PF3.

If there is a message that is displayed on the screen, PF1 gives you assistance with the message. Pressing it a second time takes you to the assistance panel for the screen you are on. Pressing it a third time takes you to the general assistance panel.

Creating a new policy database

The first thing that you need to do is to create a policy database, if you have not defined policy databases in the Customization Dialog.

Procedure

1. On the **IBM Z System Automation 4.2 Customization Dialog** Primary Menu, select option **4 Policies**. The **Policy Database Selection** panel is displayed, which lists all the available policy databases.

```
MENU  COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
                           Policy Database Selection

Action     Policy Database      Enterprise Name
****************************** Bottom of data ********************************

Command ===> _____________________________________________  SCROLL===> PAGE
```

2. Enter N or NEW on the command line. The **Create a New Policy Database** panel is displayed.

```
  COMMANDS   ACTIONS   HELP
------------------------------------------------------------------------------
                         Create a New Policy Database        Row 1 to 13 of 13

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 ********************************

Command ===> _____________________________________________________________________
```

3. On the **Create a New Policy Database** panel, fill in the following fields:
   a) In the **Policy Database Name** field, specify the policy database name, which is displayed within the Customization Dialog if you select or build the PDB. The value must be a single word but can include underscores. For example, specify TEST_PDB.
   b) In the **Enterprise Name** field, specify the name of your business or the section of it that you are going to define in the PDB. The value must be a single word but can include underscores. For example, specify TEST_SYSTEMS.
   c) In the **Data Set Name** field, specify 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 `System Operations 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` results in the data set *USER*`.TEST.PDB`, and `'AUTO.TEST.PDB'` results in the data set `AUTO.TEST.PDB`.

For the data set name, you might have to share the PDB with other members of your automation team. It is better to create the name under a shared prefix with a name that is easily specifiable to your security product ('AUTO.POLICY.TEST.PDB' for example). You need permission to create the data set before you can do so.

d) In the **Descripion** field, specify a comment for the PDB, for example the purpose of this PDB, to help you easily distinguish it with other PDBs.

e) The **Model Policy Database** field is used to select a template or an existing PDB to use as a model for your new PDB. The only one provided with SA z/OS is *EMPTY, so leave it selected.

f) The sample add-on policies at the bottom of the panel contain definitions that fully complement those in *BASE. You can select and add them to your empty policy database. Leave it alone for now.

4. When you are finished with your input, press Enter twice. (The first press validates your input, the second runs your request.) The **New Policy Database Dataset Information** panel is displayed.
5. Leave everything as it is on the **New Policy Database Dataset Information** panel and press Enter. After a few messages, you see the **Entry Type Selection** panel for your new policy database.

```
  MENU  HELP
 ------------------------------------------------------------------------------
                              Entry Type Selection
Option ===> _____________________________________________________________
                                                                More:    +
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
```

You use the panel to select the item that you want to edit. Many of these items are for advanced users. You need to focus on the most important items required now.

What to do next

Define a system (SYS), and a group (GRP) and link them together.

Creating a System

Procedure

1. On the **Entry Type Selection** panel, type 4 on the option line and press Enter. You see a panel as follows:

```
  COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
                             Entry Name Selection

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

Action      Entry Name            Short Description
****************************** Bottom of data ********************************

  No entries currently exist. Use the NEW command to create an entry.


Command ===> _____________________________________________   SCROLL===>PAGE
```

2. Enter new SYS1 on the command line and press Enter to create a system SYS1. You see a panel as follows:

```
  COMMANDS  HELP
------------------------------------------------------------------------------
                                                                More:    +
                                  Define New Entry
Command ===> ____________________________________________________________
Define new entry of type System

  Entry name . . . . . . . SYS1

  Operating system . . . . MVS     (MVS VM VSE LINUX CF ZAPP KVMIBM)
  Image/System name  . . . ___________

  Short Description  . . . ___________________________________
  Long Description 1 . . . _________________________________________________
  Long Description 2 . . . _________________________________________________
  Long Description 3 . . . _________________________________________________
  Long Description 4 . . . _________________________________________________
  Long Description 5 . . . _________________________________________________
```

3. Press Enter to complete all default entries. Edit the default entries if required, and add a description, if you want. Press PF3 to exit the panel.

   You now see a system **Policy Selection** panel, which looks as follows:

```
 ACTIONS  HELP
------------------------------------------------------------------------------
                              Policy Selection                Entry created

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

Action      Policy Name          Policy Description
__________ DESCRIPTION          Enter description
__________ SYSTEM INFO          Enter and display system information
__________ AUTOMATION SYMBOLS   Define system automation symbols (AOCCLONEx)
__________ AUTOMATION CONSOLE   Enter MVS route codes for notifications
__________ APPLICATION GROUPS   Select application groups
__________ MONITOR RESOURCES    Select monitor resources
__________ AUTOMATION TIMERS    Select timers
__________ USER E-T PAIRS       Select user entry-type pairs
__________ RESIDENT CLISTS      Select resident clists
__________ TAPE ATTENDANCE      Select tape attendance
__________ APPLICATION DEFAULTS Select application defaults
__________ SYSTEM DEFAULTS      Select system defaults
__________ MVSCOMP DEFAULTS     Select MVS component defaults
__________ MVS COMPONENT        Select MVS components
__________ NETWORK              Select network definitions
__________ AUTOMATION OPERATORS Select automation operators
__________ STATUS DISPLAY       Select status display facility details
__________ NOTIFY OPERATORS     Select notify operators
__________ OPC SYSTEM DETAILS   Select OPC system details
__________ CONTROLLER DETAILS   Select OPC controller details
__________ WORKSTATION DOMAINS  Select OPC Workstation domains

Command ===> _____________________________________________  SCROLL===>PAGE
```

4. You do not have to enter any additional data for the system, but you can review the additional definitions if you want. When you are finished, press PF3 twice to return to the **Entry Type Selection** panel.

Creating a Group

Next, create a group to hold the system.

Procedure

1. On the **Entry Type Selection** panel, enter 2 or GRP on the option line at the top and you see a panel as follows:

```
  COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
                              Entry Name Selection

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

Action      Entry Name             Short Description
****************************** Bottom of data ********************************

    No entries currently exist. Use the NEW command to create an entry.

Command ===> _____________________________________________  SCROLL===> PAGE
```

Note: You can navigate to any entry by using the fast path notation of equal sign (=) followed by the three character shorthand for the entry type. For example, =GRP entered on any Command or Option takes you to the Group dialog.

2. Enter new on the command line and press Enter. You see a panel as follows:

```
  COMMANDS  HELP
------------------------------------------------------------------------------
                                  Define New Entry

Define new entry of type Group

  Entry name . . . . . . . _____________________

  Group Type . . . . . . . SYSPLEX    (STANDARD SYSPLEX)
  ProcOps Commands . . . . NO         (YES NO)

  Short Description  . . . _______________________________
  Long Description 1 . . . ________________________________________________
  Long Description 2 . . . ________________________________________________
  Long Description 3 . . . ________________________________________________
  Long Description 4 . . . ________________________________________________
  Long Description 5 . . . ________________________________________________

Command ===> ____________________________________________________________
```

3. In the **Entry name** field, enter GROUP_ followed by the system name (for example, GROUP_SYS1) and press Enter. Then, press PF3 to close the panel. You see a panel as follows:

```
  ACTIONS  HELP
------------------------------------------------------------------------------
                              Policy Selection                 Entry created
Entry Type : Group               PolicyDB Name   : TEST_PDB
Entry Name : GROUP_SYS1          Enterprise Name : TEST_SYSTEMS

Action      Policy Name          Policy Description
__________  DESCRIPTION          Enter description
__________  GROUP INFO          Define group information
__________  SUBGROUPS           Select subgroups for group
__________  SYSTEMS             Select systems for group
__________  -------------------- -----SYSPLEX SPECIFIC POLICY-----------------
__________  SYSPLEX             Define sysplex policy
__________  APPLICATION GROUPS  Select application groups for sysplex
__________  SYSPLEX DEFAULTS    Select sysplex resource defaults for group
__________  DOMAINS             Select domains for group
__________  -------------------- -----LOCAL PAGE DATA SET POLICY--------------
__________  LOCAL PAGE DATA SET  Define local page data set recovery
__________  JOB DEFINITIONS      Define handling of jobs
__________  -------------------- -----LONG RUNNING ENQUEUE POLICY-------------
__________  JOB/ASID DEFINITIONS Define handling of long running jobs and ASID
__________  COMMAND  DEFINITIONS Define "hung" commands
__________  COMMAND  FLOODING    Define check intervals for command flooding
__________  RESOURCE DEFINITIONS Define long running enqueue resources
__________  RECOVERY OPTIONS     Define dump and miscellaneous options
__________  -------------------- ----------------------------------------------
__________  COPY                Copy data from an existing entry

Command ===> _____________________________________________  SCROLL===> PAGE
******************************* Bottom of data ********************************
```

4. Select the **SYSTEMS** policy item (type an s in front of it and press Enter) and you see a panel as follows:

```
   COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
                     Systems for Group                          Row 1 of 1

 Entry Type : Group               PolicyDB Name   : TEST_PDB
 Entry Name : GROUP_SYS1          Enterprise Name : TEST_SYSTEMS

  Action      Status            System
 _________                      SYS1
******************************* Bottom of data ********************************


Command ===> _________________________________________________  SCROLL===> PAGE
```

5. Select SYS1. The **Status** turns to **SELECTED**.
6. Press PF3 multiple times to return to the **Entry Type Selection** panel.

What to do next

Define applications and their relationships.

Adding applications

Next, you add some applications. For this exercise, you are going to create 3 applications: TSO, VTAM, and JES, because they are well-known applications. The policies that you create for them are fairly basic.

Procedure

1. Enter a 6 (or APL) on the **Entry Type Selection** panel or =APL from any other panel in the Customization Dialog. The **APL Selection** panel is displayed.

```
 COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
                             Entry Name Selection

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

Action      Entry Name             C  Short Description
****************************** Bottom of data ********************************

  No entries currently exist. Use the NEW command to create an
entry.

Command ===> _____________________________________________ SCROLL===> PAGE
```

2. You are going to create 3 applications: TSO, VTAM, and JES, starting with TSO. To create the TSO application, enter NEW TS0 on the command line. You see a panel as follows:

```
 COMMANDS  HELP
------------------------------------------------------------------------------
                                   Define New Entry
Command ===> ____________________________________________________________
                                                                More:    +
Define new entry of type Application

  Entry name . . . . . . . TSO

  Subsystem Name . . . . . ____________
  Object Type  . . . . . . INSTANCE     (CLASS INSTANCE)
  Category . . . . . . . . _________    (IBM-defined, user-defined or blank,
                                         see help)
  Subcategory  . . . . . . _________    (IBM-defined, user-defined or blank,
                                         see help)
  Job Type . . . . . . . . __________   (MVS NONMVS TRANSIENT)
  Job Name . . . . . . . . ____________________________
  Transient Rerun  . . . . __________   (YES NO)
  Scheduling Subsystem . . __________   (MSTR, JES Subsystem)
  JCL Procedure Name . . . _____________________________

  Short Description  . . . ________________________________________
  Long Description 1 . . . ________________________________________________
  Long Description 2 . . . ________________________________________________
  Long Description 3 . . . ________________________________________________
  Long Description 4 . . . ________________________________________________
  Long Description 5 . . . ________________________________________________
```

3. Do not change the default entry name of TSO. Press Enter to complete the default entries. Edit the entries, as required. While the other values default to acceptable values, it is better to enter different values:

Table 3. Application Definition Panel Options.	
Field	**Description**
Subsystem Name	The name by which automation knows the resource. It is better to keep it the same as your current job name because that is the name by which your operators know the resource. If your TSO runs as TSO, then use TSO as the automation name. If it currently runs as TCAS or anything else, then use that name. The only time that you might not want to do so is if the job name is different on each system. SA z/OS links the same application definition to multiple systems, but if you want a different automation name on each system, you either have to define separate applications for each system or you make use of the automation symbol (AOCCLONE) functionality of SA z/OS (recommended).
Object type	Leave as `INSTANCE` for now. An object type of Class defines a set of common characteristics that can be inherited by multiple instances later on.
Category	Set one of the options that are listed, as SA z/OS provides some additional automation settings and options for these resources.
Subcategory	Set if you are setting the Application Type. The Subcategory is used to differentiate between the different parts of an IMS, Db2, or CICS product for example.
Job Type	For a normal address space, the job type is set to MVS. Job type NONMVS is specified when the APL represents, for example, a UNIX System Services process or a NetView REXX script. TRANSIENTS are address spaces or jobs that shut down on their own. TRANSIENTS leave data installed and therefore it is not required to rerun them.
Job Name	For subsystems of type MVS, job name is the name of the address space, as shown in an **`MVS D A,L`** command. Typically, the job name matches your subsystem name. For subsystems of type NONMVS, it is a unique qualifier to distinguish the subsystem from other subsystems.
Transient Rerun	Valid if the subsystem is defined as a transient and indicates whether SA z/OS can ever rerun the transient before the next system IPL.

Table 3. Application Definition Panel Options. (continued)	
Field	**Description**
Scheduling Subsystem	Set to MSTR for all of the address spaces that get started before JES. Leave blank for resources that run under the default JES subsystem. If you want to specify it for subsystems that run under JES, enter the proper JES z/OS subsystem name, which is typically JES2 or JES3.

JCL Procedure Name	Name of the JCL procedure that is used to start the address space. If not specified, SA z/OS uses the following start command:
	`S jobname`
	If specified, the start command is:
	`S procname,JOBNAME=jobname`
	This command runs an address space from the procedure with the specified job name. For help information, press PF1.

4. Press Enter to confirm your input and press PF3 to close the panel. You then see a **Policy Selection** panel as follows:

```
ACTIONS  HELP
------------------------------------------------------------------------------
                              Policy Selection              Entry Selected

Entry Type : Application           PolicyDB Name   : TEST_PDB
Entry Name : TSO                   Enterprise Name : TEST_SYSTEMS

Action      Policy Name          Policy Description
_________   DESCRIPTION          Enter description
_________   UPWARD CLASS         Select a class to inherit data from
_________   APPLICATION INFO     Define application information
_________   AUTOMATION FLAGS     Define application automation flags
_________   APPLICATION SYMBOLS  Define application symbols
_________   TRIGGER              Select trigger
_________   SERVICE PERIOD       Select service period
_________   PACING GATE          Select pacing gate
_________   RELATIONSHIPS        Define relationships
_________   MESSAGES/USER DATA   Define messages and user data
_________   STARTUP              Define startup procedures
_________   SHUTDOWN             Define shutdown procedures
_________   THRESHOLDS           Define error thresholds
_________   MINOR RESOURCES      Define application minor flags and thresholds
_________   SYSTEM ASSOCIATION   Define primary and secondary associations
_________   -------------------- -----RESOURCES-------------------------------
_________   GENERATED RESOURCES  List resources generated for this entry
_________   MEMBER OF            List resources where this entry is a member
_________   -------------------- ------------------------------------------
_________   WHERE USED           List application groups linked to this entry
_________   COPY                 Copy data from an existing entry
******************************* Bottom of data ********************************

Command ===> _____________________________________________  SCROLL===> PAGE
```

5. Select the **SHUTDOWN** policy by entering an S next to **SHUTDOWN** and press Enter. The **Subsystem Shutdown Processing** panel for TSO is displayed.

```
  COMMANDS  HELP
-------------------------------------------------------------------------------
                          Subsystem Shutdown Processing
Command ===> ____________________________________________________________

Entry Type : Application           PolicyDB Name   : TEST_PDB
Entry Name : TSO                   Enterprise Name : TEST_SYSTEMS

Shutdown Pass Interval. . ________ (hh:mm:ss)

Line Commands: S/C (Cmd), R (Rep)

Cmd  Phase    Description                                  Cmd  Rep

___  INIT     Executed when shutdown is initiated
___  NORM     Executed when normal shutdown is invoked
___  IMMED    Executed when immediate shutdown is invoked
___  FORCE    Executed when force shutdown is invoked
___  FINAL    Executed after final termination message
```

This panel defines commands and replies that are issued before the application is shut down (INIT). This policy includes three increasingly aggressive shutdown command sequences (NORM, IMMED, and FORCE) and the commands that are issued after the application is shut down (FINAL).

a) For this exercise, define a NORM shutdown command. Select NORM by entering C in the **Cmd** column and press Enter. If you are going to specify a Reply, you type an R instead. The **Command Processing: SHUTNORM** panel is displayed.

```
  COMMANDS  HELP
-------------------------------------------------------------------------------
            Command Processing : SHUTNORM           Line 00000001 Col 001 075

Mixed case. . . NO_ (YES NO)

Cmd Ps AutoFn/* Command Text
___ __ ________ ____________________________________________________________
___ __ ________ ____________________________________________________________
___ __ ________ ____________________________________________________________
___ __ ________ ____________________________________________________________
___ __ ________ ____________________________________________________________
___ __ ________ ____________________________________________________________
___ __ ________ ____________________________________________________________
___ __ ________ ____________________________________________________________

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

Command ===> _____________________________________________  Scroll ===> PAGE
```

SA z/OS issues commands in passes (the **Ps** column). In the first pass, all the pass 1 commands are issued. SA z/OS then waits a while (the Shutdelay interval on the **Application Information Policy** panel) before it issues the pass 2 commands and waits again. TSO can be shut down, for example, in two passes. The first pass is an attempt to stop TSO with an **MVS STOP** command. If that does not succeed, the second pass uses the MVS CANCEL command. You define the processing on the panel as shown here.

b) For TSO, enter 1 for the pass, and MVS P TS0 as the command text.

```
Cmd Ps AutoFn/* Command Text
___ _1 ________ MVS P &SUBSJOB_____________________________________________
___ __ ________ ____________________________________________________________
___ __ ________ ____________________________________________________________
___ __ ________ ____________________________________________________________
```

Prefix the stop command with "MVS" because the commands are issued from inside the NetView. The MVS literal is required to direct the command out to the z/OS operating system.

The job name in the stop command is specified as &SUBSJOB rather than as TSO. &SUBSJOB is a symbolic value that SA z/OS substitutes with the application's job name before it issues the command. If you ever change the job name, you can change it in one place rather than locating all occurrences of the old job name.

c) Press PF3 three times to return to the **Entry Name Selection** panel.

6. Repeat this process (step 2-6) for VTAM and JES with appropriate job and procedure names and shutdown commands. For JES, you need to enter the correct Category of JES2 or JES3 and specify a Scheduling Subsystem of MSTR like as follows:

```
COMMANDS  HELP
------------------------------------------------------------------------------
                                Define New Entry
Command ===> ____________________________________________________________
                                                                More:    +
Define new entry of type Application

  Entry name . . . . . . . JES_____________

  Subsystem Name . . . . . JES_____________
  Object Type  . . . . . . INSTANCE     (CLASS INSTANCE)
  Category . . . . . . . . JES2_______  (IBM-defined, user-defined or blank,
                                         see help)
  Subcategory  . . . . . . ___________  (IBM-defined, user-defined or blank,
                                         see help)
  Job Type . . . . . . . . ___________  (MVS NONMVS TRANSIENT)
  Job Name . . . . . . . . JES________
  Transient Rerun  . . . . _____        (YES NO)
  Scheduling Subsystem . . MSTR         (MSTR, JES Subsystem)
  JCL Procedure Name . . . ______________

  Short Description  . . . _________________________________
  Long Description 1 . . . ____________________________________________________
  Long Description 2 . . . ____________________________________________________
  Long Description 3 . . . ____________________________________________________
  Long Description 4 . . . ____________________________________________________
  Long Description 5 . . . ____________________________________________________
```

7. When you finish defining these 3 applications, your **Entry Name Selection** panel now looks as follows:

```
  COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
                              Entry Name Selection           Row 1 to 3 of 3

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

Action      Entry Name             C  Short Description
__________  JES
__________  TSO
__________  VTAM
******************************** Bottom of data ********************************

Command ===> _____________________________________________ SCROLL===> PAGE
```

What to do next

Define the relationships and dependencies of the applications.

Relationships and dependencies

Next, SA z/OS would now issue all of the start commands at the same time and the starts for VTAM and TSO would fail. To avoid complications, specify the sequence in which the resources are started (and stopped).

About this task

SA z/OS uses relationships, which run from the dependent resource to the supporting resource, to control the resource start and stop sequence. Each type of relationship has a name and some can be complex. For this exercise, you are going to use one of the simple ones here: `HasParent`.

```
TSO ---- HasParent --> VTAM
VTAM --- HasParent --> JES
```

In the Customization Dialog, you always define relationships from the point of view of the dependent resource, and specify the name of the supporting resource.

Procedure

1. You start by selecting the TSO application and selecting the **RELATIONSHIPS** from its **Policy Selection** panel. The **Relationship Selection List** panel is displayed.

```
  COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
                          Relationship Selection List
Command ===> ____________________________________________________________
Entry Type : Application           PolicyDB Name   : TEST_PDB
Entry Name : TSO                   Enterprise Name : TEST_SYSTEMS

External Startup. . . ________  (INITIAL ALWAYS NEVER)
External Shutdown . . ________  (FINAL ALWAYS NEVER)

Action  #   Type             Supporting Resource               Auto    Chain
****************************** Bottom of data ******************************


   No entries currently exist. Use the NEW command to create an entry.
```

2. Type new on the command line and press Enter to create a relationship. The **Define Relationship** panel is displayed.

```
   COMMANDS  HELP
 ------------------------------------------------------------------------------
                                Define Relationship
 Command ===>
_____________________________________________________________

 Entry Type : Application           PolicyDB Name   : TEST_PDB
 Entry Name : TSO                   Enterprise Name : TEST_SYSTEMS


 Description. . . . . ________________________________________

 Relationship Type. . ________________        MAKEAVAILABLE MAKEUNAVAILABLE
                                             PREPAVAILABLE PREPUNAVAILABLE
                                             HASPARENT HASPASSIVEPARENT
                                             HASMONITOR PEEROF
                                             FORCEDOWN EXTERNALLY
 Supporting Resource.
_____________________________________
                                             Resource Name
 Sequence Number. . . ___                    Sequence Number  (1-99,blank)

 Automation . . . . . _________              ACTIVE PASSIVE
 Chaining . . . . . . _______                STRONG WEAK
 Condition  . . . . . ____________________________________________
                                             Satisfy condition
                                             (? for list of possible values)
```

3. Enter HASPARENT in the **Relationship Type** field and VTAM/APL/= in the **Supporting Resource** field. The specific notation for the supporting resource, with the equal sign (=) as a placeholder for the system name, means that the VTAM application is sharing the same system as the TSO application. If you linked the same pair of application definitions to three different systems, each copy of TSO would be dependent on the corresponding copy of VTAM.
4. Press PF3 to exit the panel. The **Relationship Selection List** panel is updated with the new relationship defined.

```
  COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
                           Relationship Selection List         Row 1 to 1 of 1
Command ===> _____________________________________________________

Entry Type : Application           PolicyDB Name   : TEST_PDB
Entry Name : TSO                   Enterprise Name : TEST_SYSTEMS

External Startup. . . _________ (INITIAL ALWAYS NEVER)
External Shutdown . . _________ (FINAL ALWAYS NEVER)

Action  #   Type             Supporting Resource               Auto    Chain
______  __  HASPARENT        VTAM/APL/=

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

5. Repeat this process to create a `HasParent` relationship from VTAM to JES.

What to do next

Group the applications and link them to the system.

Grouping applications

Although you defined the three applications and a system, they are currently not connected. The next step is to create an application group, make the applications members of the group, and then link the group to the system. You use a simple group. A simple group is a group without an automation name. It does not appear in the model that the Automation Manager uses to run the system and the applications.

Procedure

1. On the **Entry Type Selection** panel, select 5. The following panel is displayed.

```
COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
                             Entry Name Selection

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

Action      Entry Name             Short Description
****************************** Bottom of data ********************************

   No entries currently exist. Use the NEW command to create an entry.

Command ===>  _____________________________________________   SCROLL===> PAGE
```

2. Enter `new base_sys` on the command line and press Enter.

```
  COMMANDS  HELP
------------------------------------------------------------------------------
                                 Define New Entry
Command ===> ____________________________________________________________
                                                                More:    +
Define new entry of type ApplicationGroup

  Entry name . . . . . . . BASE_SYS_____________

  Type . . . . . . . . . . ______     (SYSTEM SYSPLEX)
  Nature . . . . . . . . . ______     (BASIC MOVE SERVER)
  Model  . . . . . . . . . 1          (1 2)

  Automation Name  . . . . BASE_SYS____

  Short Description  . . . ____________________________
  Long Description 1 . . . ____________________________________________
  Long Description 2 . . . ____________________________________________
  Long Description 3 . . . ____________________________________________
  Long Description 4 . . . ____________________________________________
  Long Description 5 . . . ____________________________________________
```

3. Enter SYSTEM in the **Type** field, and Basic in the **Nature** field. The **Automation Name** gets set to the entry name. Because a simple group is not created as a resource for automation, overwrite the **Automation Name** with blank spaces. Provide a description if you want. Press PF3 when you are finished and the **Policy Selection** panel is displayed.

```
  ACTIONS  HELP
------------------------------------------------------------------------------
                               Policy Selection                Entry created
Entry Type : ApplicationGroup      PolicyDB Name   : TEST_PDB
Entry Name : BASE_SYS              Enterprise Name : TEST_SYSTEMS

Action      Policy Name          Policy Description
_________   DESCRIPTION          Enter description
_________   APPLGROUP INFO       Define applicationgroup information
_________   APPLICATIONS         Select applications for system APG
_________   TRIGGER              Select trigger
_________   SERVICE PERIOD       Select service period
_________   RELATIONSHIPS        Define relationships
_________   MESSAGES/USER DATA   Define messages and user data
_________   -------------------- -----RESOURCES-----------------------------
_________   RESOURCES            Select resources and set preferences
_________   GENERATED RESOURCES  List resources generated for this entry
_________   MEMBER OF            List resources where this entry is a member
_________   -------------------- ------------------------------------------
_________   WHERE USED           List systems linked to this entry
_________   COPY                 Copy data from existing entry
******************************* Bottom of data ********************************
Command ===> _____________________________________________  SCROLL===>PAGE
```

4. Select the **APPLICATIONS** Policy.

```
COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
                    Applications for ApplicationGroup             Row 1 of 3
Entry Type : ApplicationGroup      PolicyDB Name   : TEST_PDB
Entry Name : BASE_SYS              Enterprise Name : TEST_SYTESMS

Action      Status            Application
_________                     JES
_________                     TSO
_________                     VTAM
******************************* Bottom of data ********************************

Command ===>  _____________________________________________  SCROLL===> PAGE
```

5. Select the three applications by typing an s next to each of them and pressing Enter. The Status column updates to SELECTED.

```
Entry Type : ApplicationGroup      PolicyDB Name   : TEST_PDB
Entry Name : BASE_SYS              Enterprise Name : TEST_SYSTEMS

Action      Status            Application
_________   SELECTED          JES
_________   SELECTED          TSO
_________   SELECTED          VTAM
```

6. Press PF3 to return to the **Application Group Policy Selection** panel.
7. Select the **WHERE USED** policy.

```
  COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
                    Where Used                          Row 1 to 1 of 1

Entry Type : ApplicationGroup      PolicyDB Name   : TEST_PDB
Entry Name : BASE_SYS              Enterprise Name : TEST_SYSTEMS

Action      Status            Name                      Type
_________                     SYS1                      SYS
******************************* Bottom of data ********************************

Command ===> _____________________________________________   SCROLL===>PAGE
```

8. Select SYS1 to link the application group BASE_SYS to system SYS1. Then press PF3. You see some resource creation messages displayed.

Results

You now made the link between the applications and the system, with just a simple group. Press PF4 to return to the Customization Dialog primary panel.

Reporting

The report function is used to generate reports from the policy databases. You can generate a full report of a policy database, or generate special reports which work in similar way, for example, a resource report, a symbol report, and so on. This topic describes the steps to generate a resource report and a full policy database report.

Generate a resource report

1. On the Customization Dialog primary panel, select option **3 Report** to display the **Report Selection Menu**. Except for option 1, the other reports work in similar way.

```
  MENU  HELP
------------------------------------------------------------------------------
AOFGREPK                      Report Selection Menu
Option ===> ____________________________________________________________________

  1 Report           Create Policy Database Report

  2 Unlinked         Create List of entries not linked in Policy Database
  3 ViewUnlinked     View List of unlinked entries

  4 Resources        Create List of generated Resources
  5 ViewResources    View List of generated Resources

  6 MemberList       Create List of unused members in configuration data set
  7 ViewMemberList   View List of unused members

  8 StatusMsgs       Create Status Messages Report
  9 ViewStatusMsgs   View Status Messages Report

  A Symbols          Create List of Symbols (AOCCLONEx) in Policy Database
  B ViewSymbols      View List of Symbols

  C OverrideMsgs     Create Message Override Report
  D ViewOverrideMsgs View Message Override Report

  E Runmodes         Create Runmodes Report
  F ViewRunmodes     View Runmodes Report
```

2. On the **Report Selection Menu** panel, select option **4** to generate a resource report.

```
  MENU  HELP
------------------------------------------------------------------------------
AOFGREPK                     Report Selection Menu
Option ===> 4_____________________________________________________________

  1 Report           Create Policy Database Report

  2 Unlinked         Create List of entries not linked in Policy Database
  3 ViewUnlinked     View List of unlinked entries

  4 Resources        Create List of generated Resources
  5 ViewResources    View List of generated Resources

  6 MemberList       Create List of unused members in configuration data set
  7 ViewMemberList   View List of unused members

  8 StatusMsgs       Create Status Messages Report
  9 ViewStatusMsgs   View Status Messages Report

  A Symbols          Create List of Symbols (AOCCLONEx) in Policy Database
  B ViewSymbols      View List of Symbols

  C OverrideMsgs     Create Message Override Report
  D ViewOverrideMsgs View Message Override Report

  E Runmodes         Create Runmodes Report
  F ViewRunmodes     View Runmodes Report
```

3. After the command process is displayed, press Enter to continue.

```
                         Command Progress Display

Searching resources for system SYS1
Searching resources for system SYS1
Searching resources for system SYS1
Report output stored in member "RPTRES" in data set 'YUE.YUE.TEST.PDB-
.........cont. .REP'. It can be viewed with option 5.


Press ENTER to continue
```

4. Enter 5 to view the resource report. The resources that are linked to specified systems are displayed. You see the system and the three resources that you linked to the system. There are no APG resources because you used a simple group, which does not generate a resource.

```
*************************************************************************
*
* Function                          RESOURCE REPORT
* Policy Database name              TEST_PDB
* Policy Database data set name     'YUE.YUE.TEST.PDB'
* Policy Description                test purpose
* Product Version                   IBM Z System Automatin 4.2
*
* UserId                            YUE
* Creation Date                     Monday, 5 July 2019
* Creation Time                     08:37
*
*************************************************************************
Monitor resources (MTR) for system: SYS1

Application resources (APL) for system: SYS1
  JES/APL/SYS1              JES                   via APG: BASE_SYS
  TSO/APL/SYS1              TSO                   via APG: BASE_SYS
  VTAM/APL/SYS1             VTAM                  via APG: BASE_SYS

ApplicationGroup (APG) resources for system: SYS1

Sysplex ApplicationGroup (APG) resources
```

Generate a full PDB report

1. On the **Report Selection Menu** panel, select option **1 Report**. The **Policy Database Report Parameters** panel is displayed.

2. On the **Policy Database Report Parameters** panel, rename the **Member / file name** field as you like, leave the other fields as defaulted, specify 1 on the option line, and press Enter.

```
  OPTIONS  HELP
--------------------------------------------------------------------------------
                       Policy Database Report Parameters
Option ===> 1
                                                                        More:     +
  1 Create report for a complete enterprise
  2 Create report for entry type or entry name
           Entry Type. . . . . . . . . .                        (*, ?, or type)
           Entry Name. . . . . . . . . . *                      (*, ?, or name)

Report options:
  Data set type . . . . . PDS          (PDS FS)
  PDS name  . . . . . . . YUE.YUE.TEST.PDB.REP_____________________________
  File system directory . ________________________________________________
  Member / file name  . . REPORT___________________
  Mode  . . . . . . . . . ONLINE__     (ONLINE BATCH)
  Format  . . . . . . . . FLAT__       (FLAT HTML)
  HTML file split . . . . 1__          (1 to 99 or *)

Job statement information:  (used for BATCH report)
//AOFBUILD JOB
//*
```

3. After the command process is displayed, Report Success is displayed in the upper right of the panel. Then, press PF2 to enter a new ISPF primary panel in split screen mode, enter 3 . 4 on the option line, and press Enter.
4. On the **Data Set List Utility** panel, fill in the **Dsname Level** field with the same value as the PDB data set name.

```
  OPTIONS  HELP
 ------------------------------------------------------------------------------
                       Policy Database Report Parameters    Report Successful
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
  Menu  RefList  RefMode  Utilities  Help
 ------------------------------------------------------------------------------
                           Data Set List Utility
                                                                        More:     +
   blank Display data set list              P Print data set list
       V Display VTOC information          PV Print VTOC information

Enter one or both of the parameters below:
   Dsname Level . . . YUE.YUE.TEST.PDB.REP
   Volume serial  . .

Data set list options
   Initial View               Enter "/" to select option
   1  1. Volume               /  Confirm Data Set Delete
      2. Space                /  Confirm Member Delete
      3. Attrib               /  Include Additional Qualifiers
      4. Total                /  Display Catalog Name
Option ===>
```

5. Enter **/** and press Enter to select the data set you specified. Then Press Enter again.

```
  OPTIONS  HELP
 ------------------------------------------------------------------------------
                       Policy Database Report Parameters    Report Successful
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
  Menu  Options  View  Utilities  Compilers  Help
 ------------------------------------------------------------------------------
DSLIST - Data Sets Matching YUE.YUE.TEST.PDB.REP                  Row 1 of 1

Command - Enter "/" to select action                Message           Volume
--------------------------------------------------------------------------------
 /       YUE.YUE.TEST.PDB.REP                                         AOCSM3
****************************** End of Data Set list ****************************
```

6. On the **Data Set List Actions** panel, select option **2** and press Enter.

```
                  Data Set List Actions
                                                        More:     +
Data Set: YUE.YUE.TEST.PDB.REP

DSLIST Action
2   1.  Edit                    15. Reset
    2.  View                    16. Move
    3.  Browse                  17. Copy
    4.  Member List             18. Refadd
    5.  Delete                  19. Exclude
    6.  Rename                  20. Unexclude 'NX'
    7.  Info                    21. Unexclude first 'NXF'
    8.  Short Info              22. Unexclude last 'NXL'
    9.  Print                   23. SuperC 'SC'
    10. Catalog                 24. SuperCE 'SCE'
    11. Uncatalog               25. Search-For 'SF'
```

7. Select the report you named in step 2 and press Enter to view the report. You see all the groups, systems, applications, and application groups that are defined so far in the policy database.

```
------------------------------------------------------------------------------
                              Report Selection Menu
 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
   Menu  Functions  Confirm  Utilities  Help
------------------------------------------------------------------------------
 VIEW              YUE.YUE.TEST.PDB.REP                 Row 0000001 of 0000002
            Name     Prompt       Size   Created          Changed          ID
 __s______ REPORT                  294  2018/08/06  2018/08/06 09:58:35  YUE
           **End**
```

```
 000109 ========== Applications ==========
 000110 Subsystem Name                : JES
 000111 Job Name                      : JES
 000112 Last changed by               : YUE
 000113
 000114 ========================================
 000115 TSO
 000116 ========================================
 000117
 000118 ========== Applications ==========
 000119 Subsystem Name                : TSO
 000120 Job Name                      : TSO
 000121 Last changed by               : YUE
 000122
 000123 ========== Relationships ==========
 000124 | Relationship Type | Supporting Resource |
 000125 ----------------------------------------
 000126 | HASPARENT         | VTAM/APL/=          |
 000127
 000128 ========================================
 000129 VTAM
 000130 ========================================
 000131
 000132 ========== Applications ==========
 000133 Subsystem Name                : VTAM
 000134 Job Name                      : VTAM
 000135 Last changed by               : YUE
 000136
 000137 ========== Relationships ==========
 000138 | Relationship Type | Supporting Resource |
 000139 ----------------------------------------
 000140 | HASPARENT         | JES/APL/=
|
```

Adding more systems

If you want to add extra systems, you can create the systems and link them to their Sysplex Group. Then, by linking the BASE_SYS APG to the newly created systems, all the resources within the BASE_SYS APG are created automatically for these new systems.

Procedure

1. Select option **4 Systems** on the **Entry Type Selection** panel.
2. In this exercise, you can create 2 more systems: SYS2 and SYS3. For instance, to create a new system entry for system SYS2, enter `new  SYS2`.
3. Press **Enter** to complete all the default entries. Edit the default entries if required, and add a description, if you want. Press **PF3** to exit the panel. You now see a system **Policy Selection** panel.
4. Use **APPLICATION GROUPS** policy to select the BASE_SYS APG.
5. Use **WHERE USED** policy to select their Sysplex Group.
6. Run the resources report again.

```
Application resources (APL) for system: SYS1

  JES/APL/SYS1              JES                   via APG: BASE_SYS
  TSO/APL/SYS1              TSO                   via APG: BASE_SYS
  VTAM/APL/SYS1             VTAM                  via APG: BASE_SYS

Application resources (APL) for system: SYS2

  JES/APL/SYS2              JES                   via APG: BASE_SYS
  TSO/APL/SYS2              TSO                   via APG: BASE_SYS
  VTAM/APL/SYS2             VTAM                  via APG: BASE_SYS

Application resources (APL) for system: SYS3

  JES/APL/SYS3              JES                   via APG: BASE_SYS
  TSO/APL/SYS3              TSO                   via APG: BASE_SYS
  VTAM/APL/SYS3             VTAM                  via APG: BASE_SYS
```

As you can see, it is easy to reuse existing policy elements to populate extra systems.

Cloning with symbolic values

There are times when the policy is similar but not the same on each system. So far in this example, the job name for VTAM is VTAM, but now suppose that it is NET*nn*, where *nn* is the systems subarea identifier. The following job names for VTAM are given as shown in Table 4 on page 55.

Table 4. Possible Job name values	
SYS1	NET31
SYS2	NET47
SYS3	NETT2

The subarea identifiers are deliberately used to make the values diverse.

Now, you might think that you are required to define a different VTAM application for each system and a different application group to link it to the appropriate system. Fortunately, SA z/OS supports a mechanism that is called cloning. With cloning, you specify substitutable fields in various parameters, including an application's job name, and specify different values to substitute for each system.

Procedure

1. On the **Entry Type Selection** panel, type 6 on the option line and press **Enter**. The APL Selection panel is displayed.

2. Select the application 'VTAM' and edit the **Application Information** panel for the VTAM APL and specify NET&AOCCLONE1. as the job name. Press PF3 to save the changes and exit the panel.

```
  COMMANDS  HELP
------------------------------------------------------------------------------
                          Application Information              Line 00000001

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

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 . . . . . . . . . NET&AOCCLONE1.
Transient Rerun  . . . . . ____         (YES NO)
Scheduling Subsystem . . . ____         (MSTR, JES Subsystem)
JCL Procedure Name . . . . ________________
Job Log Monitor Interval . _____        (mm:ss NONE)
Captured Messages Limit  . ___          (0 to 999)

Desired Available  . . . . __________   (ALWAYS ONDEMAND ASIS)
Restart after IPL  . . . . ________     (START NOSTART NONE)
Monitor for IPL complete . ___          (YES NO)

Start Delay  . . . . . . . ________     (time for "UP" status checks, hh:mm:ss)
Start Cycles . . . . . . . __           (start delay checks, 0 to 99)
UP Status Delay  . . . . . ________     (time to delay "UP" status, hh:mm:ss)
Restart option . . . . . . ___________  (ALWAYS ABENDONLY NEVER)
External Startup . . . . . _______      (INITIAL ALWAYS NEVER)

Command ===> _____________________________________________   Scroll===> PAGE
```

3. Enter =SYS on the command line to jump to the **System Section** panel. Select one of the systems, starting from SYS1.
4. Select the **Automation Symbols** policy and enter the correct data for the AOCCLONE1 symbol. In this exercise, enter 31 for SYS1.

```
  COMMANDS  HELP
------------------------------------------------------------------------------
                      System Automation Symbols (AOCCLONEx)
Command ===> ____________________________________________________________

Entry Type : System               PolicyDB Name   : TEST_PDB
Entry Name : SYS1                 Enterprise Name : TEST_SYSTEMS
                                                          More:     +
AOCCLONE . . . . ________
General Use:
 Description . . _____________________________________________________________
AOCCLONE1. . . . 31
 General Use:
 Description 1 . Network subarea identifier_____________________________________
AOCCLONE2. . . . ________
 General Use:
Description 2 . _____________________________________________________________
AOCCLONE3. . . . ________
 General Use:
Description 3 . _____________________________________________________________
AOCCLONE4. . . . ________
 General Use:
```

There is no reason why you are using AOCCLONE1 for the subarea identifier. If you are consistent between systems about the value that you assign to each clone identifier, you can set up the values how you want. There are a total of 36 clone values that can be assigned, that is, AOCCLONEx variables (where x is 0-9, A-Z). You need to plan carefully to avoid conflicts. Under the Enterprise Entry, there is a policy that defines your AOCCLONEx descriptions on a global basis. The dialog shows the descriptions whenever you edit a system's clone values.

5. Repeat the definitions for SYS2 and SYS3, being sure to enter the correct values.

```
Entry Type : System               PolicyDB Name   : TEST_PDB
Entry Name : SYS2                 Enterprise Name : TEST_SYSTEMS
                                                          More:     +
AOCCLONE . . . . ________
General Use:
 Description . . ____________________________________________________________
AOCCLONE1. . . . 47______
 General Use:
Description 1 . Network subarea identifier_____________________________________
```

```
Entry Type : System               PolicyDB Name   : TEST_PDB
Entry Name : SYS3                 Enterprise Name : TEST_SYSTEMS
                                                          More:     +
AOCCLONE . . . . ________
 General Use:
 Description . . ____________________________________________________________
AOCCLONE1. . . . T2______
 General Use:
Description 1 . Network subarea identifier_____________________________________
```

6. Next is verification. You cannot check the definitions with resources report because this report lists the application automation name but not its job name. You can generate the Symbol report (option A to generate and option B to view) to show your defined symbols.

```
*********************************************************************
*
-------- Symbolname ----------- |General Use
System              |Value      |Description
*
-------- AOCCLONE  ------------ |
*
-------- AOCCLONE1 ------------ |
SYS1                |31         |Network subarea identifier
SYS2                |47         |Network subarea identifier
SYS3                |T2         |Network subarea Identifier
```

   And a full report (see “Generate a full PDB report” on page 52) verifies that the job name for VTAM is correctly defined.

```
Applications
Subsystem Name      VTAM
Job Name            NET&AOCCLONE1.

Relationships
Relationship Type    Supporting Resource
HASPARENT            JES/APL/=
```

   To actually see the results of the substitution, you would have to generate a full SOCNTL data set and load the data set into a running instance of SA z/OS.

Although you now run different jobs on each of the systems, the definition you made for the applications in your dependency structure is still present. They are present because the dependencies are defined in terms of the automation names assigned to the applications rather than their job names.

Defaults, classes, and inheritance

Defaults

Defaults are a mechanism that can save you time. SA z/OS provides defaults that are hardcoded for most values. You can also specify your own set of defaults for Applications, Systems, Sysplexes, and MVS Components.

Take the Application Defaults for example. On the **Entry Type Selection** panel, select **36 ADF** (Application Defaults) entry and define a new entry.

```
  COMMANDS  HELP
------------------------------------------------------------------------------
                        Application Information Defaults
Command ===>  _____________________________________________________________

Entry Type : Application Defaults  PolicyDB Name   : TEST_PDB
Entry Name : MY_DEFAULTS           Enterprise Name : TEST_SYSTEMS

 Captured Messages Limit. . ___         (0 to 999)
 Monitor Routine  . . . . . ________    (name NONE)
Startup information:
 Restart after IPL  . . . . _______     (START NOSTART)
 Start Delay  . . . . . . . ________    (time for "UP" status checks, hh:mm:ss)
 Start Cycles . . . . . . . __          (start delay checks, 0 to 99)
 Restart Option . . . . . . _________   (ALWAYS ABENDONLY NEVER)
Shutdown information:
 Shutdown Pass Interval . . ________    (hh:mm:ss)
 Cleanup Delay  . . . . . . ________    (hh:mm:ss)
```

The defaults are mainly for the timing intervals between various actions that the agent takes during the automation of the resource.

Classes and inheritance

A powerful technique available with the Customization Dialog is that of classes and inheritance. For example, if you are running five Db2 subsystems, you can create a single Db2 Application Class, which holds all of the complicated Db2 automation policies. Then, you link it as the Upward Class of the five Application instances that represent your Db2 subsystem. You do not have to type in and maintain your potentially complex Db2 automation five times.

The following information shows the procedure of this example.

1. On the **Entry Type Selection** panel, select **6 APL**.
2. Enter new DB2_CLASS to create a new APL entry type with an **Object Type** of CLASS.

```
  COMMANDS  HELP
------------------------------------------------------------------------------
                                  Define New Entry
Command ===> ____________________________________________________________

Define new entry of type Application

  Entry name . . . . . . . DB2_CLASS

  Subsystem Name . . . . . _____________
  Object Type  . . . . . . CLASS        (CLASS INSTANCE)
  Category . . . . . . . . __________   (IBM-defined, user-defined or blank,
                                         see help)
  Subcategory  . . . . . . __________   (IBM-defined, user-defined or blank,
                                         see help)
```

3. For the DB2_CLASS entry, fill out the policies only to the degree that is common (or shared) among all Db2 subsystem instances that you plan to create. Follow the same steps that you did when creating other APL entries before.
4. Define the Db2 subsystems applications. Set **Object Type** to INSTANCE.

```
  COMMANDS  HELP
------------------------------------------------------------------------------
                                  Define New Entry
Command ===> ____________________________________________________________

Define new entry of type Application

  Entry name . . . . . . . DB2_T100MSTR_____

  Subsystem Name . . . . . T100MSTR_____
  Object Type  . . . . . . INSTANCE     (CLASS INSTANCE)
  Category . . . . . . . . __________   (IBM-defined, user-defined or blank,
                                         see help)
  Subcategory  . . . . . . __________   (IBM-defined, user-defined or blank,
                                         see help)
  Job Type . . . . . . . . _________    (MVS NONMVS TRANSIENT)
  Job Name . . . . . . . . T100MSTR
  Transient Rerun  . . . . ___          (YES NO)
```

5. In the applications' **UPWARD CLASS** policy, select the Class you defined in step 1. With this specification, the application that is linked to the class inherits all of the Class's policies.

```
 COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
                            Upward Link to Class                Row 1 of 1
Command ===>  _____________________________________________   SCROLL===> PAGE

Entry Type : Application           PolicyDB Name   : TEST_PDB
Entry Name : DB2_T100MSTR          Enterprise Name : TEST_SYSTEMS

Action      Status     Entry Name            Type        Upward Class Link
________    SELECTED   DB2_CLASS
****************************** Bottom of data ********************************
```

Adapt the sample policies

When you are using the Customization Dialog tutorial, you notice there are many tasks that are involved in creating your own automation policy from scratch. It's much easier to start with sample policies that come with SA z/OS.

You now use the *BASE add-on or sample policy and customize the key components that are required for this installation. You can start with a new policy database by following the instructions that are given in "A new policy database" on page 59. Alternatively, you can start with the policy database you created, installed, and verified as described in "Start SA z/OS for the first time" on page 18.

A new policy database

Procedure

1. Create a new policy database, starting from the Customization Dialog primary panel. Select option **4** for the **Policies** choice.

```
  MENU  OPTIONS  HELP
------------------------------------------------------------------------------
           IBM Z System Automation 4.2 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 . . . TEST_PDB

                  Licensed Materials - Property of IBM
                    ©  Copyright IBM Corp. 1990, 2019
```

2. Type new SA_FULL on the command line and press Enter. You can use whatever name you want for your enterprise.
3. Complete the **Data Set Name** field. Remember you might not want to use a personal HLQ if the data set is to be shared between several updaters.

```
  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 . . SA_FULL__________
   Enterprise Name. . . . . SA_FULL__________
   Data Set Name. . . . . . SAPDB_____________________________________________
   Desciption . . . . . . . _____________________________________________________

   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
________               *PROCOP
________               *SAPSRV
________               *TBSM
________               *ZWS
******************************* Bottom of data ********************************
```

4. Enter c in the **Action** field against the *BASE policy to customize it.

```
 COMMANDS  ACTIONS  HELP
------------------------------------------------------------------------------
                     Select Add-on Policy Components        Row 1 to 14 of 14

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    Base Communication Services (USS)
______ SELECTED    File Transfer Protocol Daemon (FTPD)
______ SELECTED    IBM HTTP Server
______ SELECTED    Job Entry Subsystem 2
______ SELECTED    Job Entry Subsystem 3
______ SELECTED    Network File System Server and Client (NFS)
______ SELECTED    Secure Shell Daemon (SSHD)
______ SELECTED    TELNET Server for 3270 (TN3270)
______ SELECTED    Dynamic Virtual IP Addressing (DVIPA)
****************************** Bottom of data ********************************
```

5. Customize the selections. If you don't need a selection, enter `m` in the **Action** field and press Enter to remove it, for example, Job Entry Subsystem 3.
6. After customization, press PF3 to close this panel.
7. Press Enter and the data set allocation panel comes up.

```
 COMMANDS  HELP
------------------------------------------------------------------------------
                    New Policy Database Dataset Information

New Policy Database :  SA_FULL
Attributes to be used for allocation of the new data set:
   Managed storage. . . . . YES______   YES      NO
   Management class . . . . STANDARD_  Blank for default management class   *
   Storage class. . . . . . SMS_____  Blank for default storage class      *
    Volume serial. . . . .  _______    Blank for authorized default volume
   Data class . . . . . . . ________   Blank for default data class         *
     Space units. . . . . . CYLINDERS   CYLS TRKS BLKS KB MB
     Primary quantity . . . 60_         1 to 999 - In above units
     Secondary quantity . . 12_         0 to 999 - In above units
     Directory blocks . . . 20_         1 to 999
     Record format. . . . : FB
     Record length. . . . : 80
     Block size . . . . . . 32720
     Data Set Name type . . PDS_____    LIBRARY  PDS
     Device Type. . . . . . 3390____
  * Used only if Managed storage = YES

Command
===>_____________________________________________________________________________
```

8. When you're ready, press Enter to begin creating a policy database. You see some progress messages as your database is created and the sample policies are copied over.

```
  COMMANDS  HELP
- ┌──────────────────────────────────────────────────────────────────────┐
  │                          Command Progress Display                    │
  │                                                                      │
  │ Source : 'SYS18207.T045948.RA000.YUE.R0100315'                       │
N │ Target : 'USER1.SAPDB'                                               │
A │ Copy complete                                                        │
  │ Adding entry for "SA_FULL" to list of Databases                      │
  │ Rename of entries associated with Enterprise name                    │
  │ Rename complete                                                      │
  │ Open the selected add-on policies.                                   │
  │ Analyze add-on policy '*BASE'.                                       │
  │ Read data from add-on policy '*BASE'.                                │
  │ Opening tables.                                                      │
  │ Progress : ____________________ 36 %                                 │
  │                                                                      │
  │                                                                      │
  │ PF 1=HELP     2=SPLIT    3=END      4=RETURN   5=RFIND    6=RCHANGE  │
  │ PF 7=UP       8=DOWN     9=SWAP    10=LEFT    11=RIGHT                │
  └──────────────────────────────────────────────────────────────────────┘
     Data Set Name type . . PDS         LIBRARY  PDS
```

9. On completion, you are on the **Entry Type Selection** panel for your new policy database.

Results

Look at the APLs and APGs that are defined (options 5 and 6). As you can see, there are quite a few of them. Many of the APLs likely have the wrong job names for your systems.

At this stage, the default system name might need to be renamed. You also need to prepare the recovery flag for certain MVSESA resources.

Adding sample policies

Before you begin

Start from the Customization Dialog primary panel with your policy database selected.

```
  MENU  OPTIONS  HELP
------------------------------------------------------------------------------
           IBM Z System Automation 4.2 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 . . . SA_FULL_____________

                   Licensed Materials - Property of IBM
                     ©  Copyright IBM Corp. 1990, 2017
```

Procedure

1. Select option **5 Data Management** > option **2 Import from Add-on**. The **Import Add-on Policies** panel is displayed.

```
ACTIONS   HELP
 ------------------------------------------------------------------------------
                              Import Add-on Policies                Row 1 of 1

 1 Import selected add-on policies
 2 View import report

 Current Policy Database: SA_FULL
 Add-on policies to be added to the current policy database:
 Action     Status      Add-on Policy         Customizable
 ________               *BASE                 YES
 ________               *CICS
 ________               *DB2
 ________               *E2E                  YES
 ________               *GDPS
 ________               *HYPERSWAP
 ________               *IBMCOMP              YES
 ________               *IMS
 ________               *ITM                  YES
 ________               *PROCOP
 ________               *SAPSRV
 ________               *TBSM
 ________               *ZWS
 ******************************* Bottom of data ********************************

Option ===> ____________________________________________________________________
```

2. Enter an s in the **Action** column to select the *BASE policy and press Enter.
3. Enter 1 on the option line and press Enter. After you see some processing progress messages, you see this panel.

```
OPTIONS  ACTIONS  HELP
 ------------------------------------------------------------------------------
                       Entries of selected Add-on Policies    Row 1 to 13 of 84

 1 Modify data        2 Check data         3 Run import         4 View report

 Action Entry Name           Type C D Short Description
 ______ SYSPLEX1             GRP    Y SA Sample Sysplex
 ______ SYS1                 SYS    Y System 1 of the SA Sample Sysplex
 ______ SYS2                 SYS    Y System 2 of the SA Sample Sysplex
 ______ SYS3                 SYS    Y System 3 of the SA Sample Sysplex
 ______ AM_X                 APG    Y Automation Manager Group for Sysplex
 ______ BASE_SYS             APG    Y Base z/OS, Network and Automation
 ______ BASE_USS             APG    Y Base USS resources
 ______ FTP_DAEMON           APG    Y File Transfer Protocol Daemon
 ______ MYSERV1              APG    Y Sample Server Application #1
 ______ MYSERV1_X            APG    Y Virtual IP Adressing Server
 ______ NFS_CLNT             APG    Y NFS client
 ______ NFS_SERV_X           APG    Y NFS Server
 ______ SSH_DAEMON           APG    Y Secure Shell Daemon

Option ===> ____________________________________________________   SCROLL===> PAGE
```

This panel can be used to import policies into your existing policy database selectively. You now import all the policies that you think you need for your initial automation. It might be no more than the starter set that you put together for the verification test or it can be every policy in the sample. You can return to this panel to import more policies at any time.

On this panel, you can do the following things:

- Remove policies that you do not want to import by entering an M against them. **Important:** If you remove something you actually want, there is no way to restore it on this import run.
- Change entry names by entering an R against them and enter the new names.
- Change the short descriptions by overtyping them.
- Request a global substitution of 1 string for another by entering 1 on the option line.
- Check the data by entering 2 on the option line.
- Import the policies by entering 3 on the option line.
- View the report of your last import run by entering 4 on the option line.

It is not sufficient to change the entry name to differentiate it from previous instances of the policy that might be imported. You need to perform a global substitution. For example, changing the entry name of the SYS1 entry to SYS2 does not import it a second time. However, performing a global substitution of SYS1 with SYS2 does import it.

Note: Global substitution works by affecting all policies that are still in the list. Use with caution and with small groups of policies or you might see unexpected consequences.

4. Press PF3 to exit without doing anything.

   Any changes that were not imported are lost.

Bulk update

Although you could go through each of the defined applications (APL) in sequence, open up its Application Information policy item and update its job name. There is an alternative faster way of editing application policy information, by using the Update via File function.

Export

Press PF3 to go to the Customization Dialog primary panel. Select option **5 Data Management** > option **3 Update via File**. The **Policy Database Update Selection** panel is displayed.

Specify APL as the selection criteria and give it an output data set name, for example, `SA.FULL.FLATFILE`. Specify 1 on the option line and press Enter.

```
   MENU  HELP
 ------------------------------------------------------------------------------
                     Policy Database Update Selection
Option ===> 1_____________________________________________________________

   1 Write selected data from Policy Database to file
        Entry Type . . . . . . . APL            (? or type)
        Output File Name . . . . SA.FULL.FLATFILE_____________________________

   2 Perform syntax check for data in file
   3 Update Policy Database with data from file
        Input File Name. . . . . _____________________________
        Mode . . . . . . . . . . ONLINE               (ONLINE BATCH)

   4 View write / update report

   5 Edit output file

   6 Append NEW/DEL templates to output file

 Job statement information:  (used for BATCH update)
 //AOFBUILD JOB_____________________________________________________________
 //*_____________________________________________________________________
 //*_____________________________________________________________________
```

If the output data set name does not exist, it is allocated when the file update starts. If a name is not specified, a default data set name is used, which is the PDB data set name with `.UPD` appended as low-level qualifier.

You now have the **Entry Name Selection** panel, showing all the APLs in the sample policy.

```
  COMMANDS  ACTIONS  HELP
------------------------------------------------------------------------------
                  Entry Name Selection                      Row 1 to 15 of 57

Action   Status   Entry Name           Short Description
_______           AM                   SA Automation Manager
_______           AM2                  Spare SA 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_DVIPA              Class for common DVIPA definitions
_______           C_JES2               Class for Job Entry Subsystem 2
_______           C_USS_APPL           Class for USS Application
_______           C_USS_FILE           Class for USS Files
_______           C_USS_PORT           Class for USS Port
_______           CRON                 Time-based job scheduler in Unix like OS
_______           DLF                  Data Lookaside Facility
_______           DSIRQJOB             NetView JES-JobID-
Requestor

Command ===> ____________________________________________     SCROLL===>PAGE
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
```

Select all of the APLs that are not classes (so, ignore the ones that start with a C_ prefix). Type `s *` on the command line and press **Enter**. Then, enter an `m` against each of the C_ APLs and press **Enter**. Scroll down to get all of the C_ APLs.

You end up with the following panel:

```
  COMMANDS  ACTIONS  HELP
------------------------------------------------------------------------------
                  Entry Name Selection                        Row 25 of 105
Command ===> ________________________________________________  SCROLL===>PAGE

Action   Status   Entry Name           Short Description
         SELECTED AM                   SA Automation Manager
         SELECTED AM2                  Spare SA Automation Manager
         SELECTED APPC                 Advanced Peer-to-Peer Communication
         SELECTED ASCH                 APPC Scheduler
         SELECTED BLSJPRMI             Build SNAP Tables for IPCS
                  C_AM                 Class for Automation Manager Definitions
                  C_APPL               Class for general APL definitions
                  C_DVIPA              Class for common DVIPA definitions
                  C_JES2               Class for Job Entry Subsystem 2
                  C_USS_APPL           Class for USS Application
                  C_USS_FILE           Class for USS Files
                  C_USS_PORT           Class for USS Port
         SELECTED CRON                 Time-based job scheduler in Unix like OS
         SELECTED DLF                  Data Lookaside Facility
         SELECTED DSIRQJOB             NetView JES-JobID-Requestor
```

Press Enter and you see the **Policy Selection** panel. For now, select the DESCRIPTION, APPLICATION INFO, and USS CONTROL policy data. Enter an S against them to select them and press Enter.

```
  COMMANDS  ACTIONS  HELP
------------------------------------------------------------------------------
                        Policy Selection                    Row 1 to 13 of 24
Select one or more policies to be written to file:

Action    Status     Policy Name
_______   SELECTED   DESCRIPTION
_______              UPWARD CLASS
_______   SELECTED   APPLICATION INFO
_______              AUTOMATION FLAGS
_______              APPLICATION SYMBOLS
_______              RELATIONSHIPS
_______              MESSAGES/USER DATA
_______              STARTUP
_______              SHUTDOWN
_______              THRESHOLDS
_______              MINOR RESOURCES
_______              CICS CONTROL
_______              DB2 CONTROL
_______              OPC CONTROL
_______              IMS CONTROL
_______   SELECTED   USS CONTROL
_______              JES2 SPOOLFULL CNTL
_______              JES2 SPOOLFULL CMDS
_______              JES2 SPOOLSHORT CNTL
_______              JES2 SPOOLSHORT CMDS
_______              JES2 DRAIN CMDS
_______              JES3 SPOOLFULL CMDS
_______              WHERE USED
```

Press Enter again after you selected them and you see a Confirmation Panel. The panel, when you scroll it, contains the names of all of the APLs that you want to export.

```
  COMMANDS  HELP
------------------------------------------------------------------------------
                        Write to File Confirmation           Row 1 to 11 of 50

The following entries have been selected to write data to file.
Press ENTER to write data to file.
Press CANCEL or END to cancel writing data.

Selected Names        Description
AM                    SA Automation Manager
AM2                   Spare SA Automation Manager
APPC                  Advanced Peer-to-Peer Communication
ASCH                  APPC Scheduler
BLSJPRMI              Build SNAP Tables for IPCS
CRON                  Time-based job scheduler in Unix like OS
DLF                   Data Lookaside Facility
DSIRQJOB              NetView JES-JobID-Requestor
DVIPA1                Dynamic Virtual IP Address
FFST                  First Failure Support Technology
FTP_PORT              Port which is used by FTP daemon

Command ===> _____________________________________________  SCROLL===>PAGE
```

Press Enter and a progress dialog is displayed.

```
  COMMANDS  HELP
- ┌──────────────────────────────────────────────────────────────────────┐
  │                          Command Progress Display                    │
  │                                                                      │
  │ Writing data to file for APL CICS1TGD.                               │
T │ Writing data to file for APL CICS1TGM.                               │
P │ Writing data to file for APL CICS1TOR.                               │
P │ Writing data to file for APL CMDRCVR.                                │
  │ Writing data to file for APL CPSMCMAS.                               │
  │ Writing data to file for APL CPSMLIST.                               │
  │ Writing data to file for APL CPSMWUI.                                │
  │ Writing data to file for APL CRON.                                   │
  │ Writing data to file for APL DB21ADMT.                               │
  │ Writing data to file for APL DB21DBM1.                               │
  │                                                                      │
  │                                                                      │
  │                                                                      │
  │ F1=HELP     2=SPLIT    3=END      4=RETURN   5=RFIND    6=RCHANGE    │
  │ F7=UP       8=DOWN     9=SWAP    10=LEFT    11=RIGHT                 │
  └──────────────────────────────────────────────────────────────────────┘
```

On completion, you return to the **Policy Database Update Selection** panel:

```
  MENU  HELP
 ------------------------------------------------------------------------------
                     Policy Database Update Selection      Data written to file
Option ===>  ____________________________________________________

   1 Write selected data from Policy Database to file
        Entry Type . . . . . . . APL              (? or type)
        Output File Name . . . . SA.FULL.FLATFILE_____________

   2 Perform syntax check for data in file
   3 Update Policy Database with data from file
        Input File Name. . . . . SA.FULL.FLATFILE_____________
        Mode . . . . . . . . . . ONLINE               (ONLINE BATCH)

   4 View write / update report

   5 Edit output file

   6 Append NEW/DEL templates to output file
```

You notice that the name of the output file is completed under option 3 > **Input File Name**. Now, do not use the Customization Dialog to edit any of the APL data, which are written to file. The import step that you intend to make issues a warning if those data sets are edited through the dialog between when you exported the data and when you re-import it. And if you confirm running the import, then it overwrites the intermediate changes in the PDB.

Update

You might want to do some research and planning before you start the update step. Find out all your job names and procedure names. Find out your UNIX System Services paths, ports, and files. If you want to use AOCCLONE values to share definitions between multiple systems, plan the clone values that you are going to use. Then, work out the job names with the clone substitutions specified.

Select option **5** to edit the data in the file:

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
_____________________________________________________________________
EDIT       YUE.SA.FULL.FLATFILE                          Columns 00001 00072

****** ***************************** Top of Data ******************************
000001 * UserId                    YUE
000002 * PolicyDB Name             SA_FULL
000003 * PolicyDB Data Set         YUE.SA.FULL.PDB
000004 * Version                   4.2
000005 * Creation Date             Monday,26 July 2019
000006 * Creation Time             08:31:56
000007
000008 FILE CREATION TIMESTAMP     20190726092759
000009
000010 UPD APL                            - AM
000011   DESCRIPTION                      ( field replacement )
000012     Short Description              : SA Automation Manager
000013     Long Description 1             : The AM runs on multiple systems of the sysplex.
000014     Long Description 2             : One of them is operating as the Primary AM.
000015     Long Description 3             :

Command ===> _____________________________________________  Scroll===> PAGE
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=Rfind     F6=Rchange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

The file display is a standard ISPF edit session on the data set holding the exported data. The data in the file is case-sensitive and the indentations on the left side are needed to indicate which fields belong together, so the rule here is only to change the minimum you have to.

For each Application in the file (starting with a UPD APL line), scroll down to its job name. Type it over with the correct job name for your system. For example, if the job name of your Automation Manager is SAAM, you would find the current job name:

```
000032     Monitor Interval             :
000033     Application Type             :
000034     Subcategory                  :
000035     Subsystem Name               : AM
000036     Job Type                     :
000037     Job Name                     : AM
000038     Transient Rerun              :
000039     Scheduling Subsystem         :
000040     JCL Procedure Name           :
000041     Job Log Monitor Interval     :
```

and type it over with the correct job name:

```
000032     Monitor Interval             :
000033     Application Type             :
000034     Subcategory                  :
000035     Subsystem Name               : AM
000036     Job Type                     :
000037     Job Name                     : SAAM
000038     Transient Rerun              :
000039     Scheduling Subsystem         :
000040     JCL Procedure Name           :
000041     Job Log Monitor Interval     :
```

You can update the contents of other fields, for example, the JCL Procedure Name. While it is suggested that the SUBSYSTEM name match the job name, changing it here means you have to make a number of corresponding changes to Application relationships and other definitions. It is simpler to leave it for now.

If you are using AOCCLONE values, remember to type the job name with the &AOCCLONE. value specified, not a resolved value:

```
000032     Monitor Interval             :
000033     Application Type             :
000034     Subtype                      :
000035     Subsystem Name               : VTAM
000036     Job Type                     :
000037     Job Name                     : NET&AOCCLONE1.
000038     Transient Rerun              :
000039     Scheduling Subsystem         :
000040     JCL Procedure Name           :
000041     Job Log Monitor Interval     :
```

For APLs where your systems run multiple instances, IMS, DB2®, CICS®, WebSphere®, set the job names for just one set of instances. We duplicate and rename them later using a different mechanism.

The UNIX System Services Applications have a USS CONTROL section:

```
  USS CONTROL                   ( policy replacement )
    User ID                     : WEBSRV
    Monitoring Command          :
    Process Command/Path        : /IHS_RUNTIME_DIRECTORY/bin/httpd
    with Filter                 : -d /IHS_RUNTIME_DIRECTORY -k start -f
    File Name                   :
    IP Stack                    :
    Port Number                 :
```

In addition to correcting the job name, you need to check the UNIX System Services data as well. Blank is acceptable. It defaults to a value from a class. But you can explicitly specify it; however if it is wrong, you need to fix it.

Interruptions

If you need to interrupt your work, press PF3 to save the file and you are returned to the **Policy Database Update Selection** screen. Do not attempt to import your data until you are finished. Doing so means you get a warning for ALTERED data in the PDB and you have to confirm the import. If you import the data when you interrupt the work, then it is recommended to re-export it before you make any further updates. You can exit the Customization Dialog and log off from TSO if you want.

To resume editing the data, choose option **5** again. Or open the data set through the normal ISPF editor. You can also print it or download it. But be careful with downloading it to edit it on your workstation as code page mismatches cause problems. If you did download it to edit it, you must upload it into the same data set before you can import your updates.

Import

When you make all your updates, return to the **Policy Database Update Selection** screen and choose option 2. The syntax and alignment of the data file is validated.

```
  MENU  HELP
- ┌──────────────────────────────────────────────────────────────────┐
  │                          Command Progress Display                │
O │                                                                  │
  │ Checking syntax of function UPD for APL TWSSVE2E.                │
  │ Checking syntax of function UPD for APL TWSSVJSC.                │
  │ Checking syntax of function UPD for APL TWSSVPIF.                │
  │ Checking syntax of function UPD for APL TWSTRKR.                 │
  │ Checking syntax of function UPD for APL VLF.                     │
  │ Checking syntax of function UPD for APL VTAM.                    │
  │ Checking syntax of function UPD for APL WEBSRV.                  │
  │ Checking syntax of function UPD for APL ZFS.                     │
  │ SUMMARY       : Check completed. 0 syntax error(s) and 0 altered │
  │ ..........cont. object(s) detected.                              │
  │                                                                  │
  │ Press ENTER to continue                                          │
  │                                                                  │
  │ F1=HELP     2=SPLIT    3=END      4=RETURN   5=RFIND    6=RCHANGE │
  │ F7=UP       8=DOWN     9=SWAP    10=LEFT    11=RIGHT              │
  └──────────────────────────────────────────────────────────────────┘
```

The "0 altered objects" means that none of the APLs are modified in the PDB since the data are written to file. There are no updates to the automation policy.

The "0 syntax error(s)" means that there are no syntax errors. If any error is found, use option **4 View write / update report** to find what the error is and then fix it using option **5 Edit output file**.

Pressing Enter brings you back to the **Policy Database Update Selection** panel. To actually write the data from the file to the PDB, choose option **3**.

You get another progress dialog where it also runs the syntax check (and stops if it finds any problems) after which it then applies your policy updates.

When it is finished, it returns you to the **Policy Database Update Selection** screen. You can view the Update report with option **4** and then press PF3 to return to the Customization Dialog primary panel. The APLs now have the right job names for your system. You can check by viewing the APL definitions.

Setting up your systems

The sample policies come with 3 systems called SYS1, SYS2, and SYS3, which are all linked to one sample Group called SYSPLEX1. You now need to adapt these systems and the Group.

If you have a look at the policies that are attached to the SYS1, SYS2, and SYS3 system, you note that many of them have data that is provided. There are linked Application Groups, defined defaults, lined MVS Component automation definitions, and so on.

It is assumed that the actual system names are TST1, TST2, and TST3.

First, rename the SYS1 entry. On the **Entry Name Selection** screen for Systems, enter RENAME against the SYS1 system:

```
  COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
                           Entry Name Selection                Row 1 of 1
Command ===> _____________________________________________  SCROLL===>PAGE

Entry Type : System               PolicyDB Name   : SA_FULL
                                  Enterprise Name : SA_FULL
Action      Entry Name            Short Description
RENAME__    SYS1                  System 1 of the SA Sample Sysplex
________    SYS1                  System 1 of the SA Sample Sysplex
________    SYS1                  System 1 of the SA Sample Sysplex
****************************** Bottom of data ********************************
```

Press Enter and you are prompted for the new name:

```
ion      Entry Name            Short Description
AME      SYS1                  System 1 of the SA Sample Sysplex
***  ┌──────────────────────────────────────────────────┐ **
     │                      Entry Rename                │
     │                                                  │
     │ Description : System 1 of the SA Sample Sysplex  │
     │ Old Name    : SYS1                               │
     │                                                  │
     │ New Name  . . _____________________              │
     │                                                  │
     │ Press ENTER to rename member.                    │
     │ Press CANCEL to cancel rename.                   │
     │  F1=Help    F2=Split   F3=End     F9=Swap   F12=Cancel │
     └──────────────────────────────────────────────────┘
```

Enter the name of the first system you want to produce (TST1) and press Enter again. A progress dialog appears and when completed you are back on the **Entry Name Selection** panel with the system renamed:

```
  COMMANDS  ACTIONS  VIEW  HELP
------------------------------------------------------------------------------
                            Entry Name Selection               Policy saved
Command ===> _____________________________________________   SCROLL===>PAGE
Entry Type : System                PolicyDB Name   : SA_FULL
                                   Enterprise Name : SA_FULL
Action      Entry Name             Short Description
_________   TST1                   System 1 of the SA Sample Sysplex
****************************** Bottom of data ********************************
```

Now, the Entry name (the name of the policy element that defines the system) is changed. To update the SYSNAME for the system, select it and edit its SYSTEM INFO policy:

```
  COMMANDS  HELP
------------------------------------------------------------------------------
                            System Information
Command ===>
_____________________________________________________________


Entry Type : System                PolicyDB Name   : SA_FULL
Entry Name : TST1                  Enterprise Name : SA_FULL
                                                             More:     +
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)
```

Type over the old SYS1 system name with TST1 to complete the rename.

If you are using JES3, you also want to type over the **PRIMARY JES** field with the subsystem name (automation name) of your JES3 Application. The default is JES3.

It is recommended when you change all relationships from JES2 to JES3 of the shipped sample policies you can use the INGEFUJ3 member in the SINGSAMP library to update the policy via file.

Now press PF3 and you are finished.

Then set up TST2 and TST3 following the process of TST1.

Sysplex Group (GRP)

The next step is to rename your Sysplex Group.

The provided sample group is called SYSPLEX1. Rename it to match your real group, GROUP_SYS in this case.

Then, edit its policies and update its description and its **Sysplex Policy Definition** item:

```
 COMMANDS  HELP
------------------------------------------------------------------------------
                          Sysplex Policy Definition
Command ===> ____________________________________________________________

Entry Type : Group                 PolicyDB Name   : SA_FULL
Entry Name : GROUP_SYS             Enterprise Name : SA_FULL

Sysplex Name. . . . . . . . . . . .
TSTPLEX_
Temporary Data Set HLQ. . . . . . . _____________________
                                      Data set HLQ (max. 17 chars)
Started Task Job Name . . . . . . . _____________________
Couple Data Set HLQ . . . . . . . . _____________________

CDS type    Alternate volumes
```

Set the Sysplex Name to the correct value.

Complete the replication

Replicate samples until you have enough definitions for all of the applications you want to run on the system.

Review

To have a look at what you created so far, go look at the resource report. You see the three systems, each with resources that are attached to them.

You need to go through the resource report in some detail and identify resources that are linked to systems that they do not run on. For example, some of your test systems may not run Db2, CICS, IMS , and ZWS. After you identify these systems, go into the APPLICATION GROUP policies for the systems and unlink the appropriate groups from the systems.

Regenerate the resource report to verify your updates. For details of generating reports, see “Reporting” on page 51.

Building your policy

Before you can distribute your policy to your target system, you need to build it into a staging data set.

As it is the first time to build this policy, you first need to allocate the staging data set. Outside of the Customization Dialog, use the ISPF 3.2 dialog. You can use PF2 to start a split screen session to do it:

```
  Menu  RefList  Utilities  Help
------------------------------------------------------------------------------
                              Data Set Utility

    A Allocate new data set                C Catalog data set
    R Rename entire data set               U Uncatalog data set
    D Delete entire data set               S Short data set information
blank Data set information                 V VSAM Utilities

ISPF Library:
   Project  . . ___________      Enter "/" to select option
   Group  . . . ___________      /  Confirm Data Set Delete
   Type . . . . ___________

Other Partitioned, Sequential or VSAM Data Set:
   Name  . . . . . . . sa.full.socntl_____________________________________
   Volume Serial . . . _________ (If not cataloged, required for option "C")
   Data Set Password  . . _________ (If password protected)

Option ===> a_____________________________________________________________
```

The convention is to allocate it with the same name as the Policy Database data set, but with an SOCNTL suffix rather than a PDB suffix. Your system programmers might require it to be allocated under a different name.

The data set is a standard FB80 PDS:

```
Data Set Name  . . . : SA.FULL.SOCNTL
Management class . . . STANDARD_____  (Blank for default management class)
Storage class  . . . . SMS__________  (Blank for default storage class)
 Volume serial . . . . AOCSM4        (Blank for system default volume) **
 Device type . . . . . ______        (Generic unit or device address) **
Data class . . . . . . _________     (Blank for default data class)
 Space units . . . . . TRACK         (BLKS, TRKS, CYLS, KB, MB, BYTES
                                      or RECORDS)
 Average record unit   _             (M, K, or U)
 Primary quantity  . . 60____        (In above units)
 Secondary quantity    12____        (In above units)
 Directory blocks  . . 0____        (Zero for sequential data set) *
 Record format . . . . FB__
 Record length . . . . 80_____
 Block size  . . . . . 32720__
 Data set name type    PDS______     (LIBRARY, HFS, PDS, LARGE, BASIC, *
                                      EXTREQ, EXTPREF or blank)
Extended Attributes   ___________   (NO, OPT or blank)
 Expiration date . . . ___________   (YY/MM/DD, YYYY/MM/DD
Enter "/" to select option           YY.DDD, YYYY.DDD in Julian
form
 Allocate Multiple Volumes _        DDDD for retention period in days
                                      or blank)
```

3 Cylinders are ample for your initial allocation.

Return to the SA z/OS Dialog after you allocated it.

Select option 2 (Build) from the Customization Dialog primary panel. The build dialog is displayed:

```
  MENU  HELP
------------------------------------------------------------------------------
                              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. . . . . . . . ___                   (*, ?, or type)
      Entry Name. . . . . . . . ____________________  (*, ?, or name)
  4 View build report

Build options:
  Output Data Set . . . . ________________________________________
  Mode. . . . . . . . . . ONLINE____  (ONLINE BATCH)
  Type. . . . . . . . . . ALL_______  (MODIFIED ALL)
  Configuration . . . . . NORMAL____  (NORMAL ALTERNATE TERTIARY)

Job statement information:  (used for BATCH build)
//AOFBUILD_JOB___________________________________________________________
//*______________________________________________________________________
//*______________________________________________________________________
```

Specify your output data set name and ask for a full build (option 1).

You see a process dialog with messages that indicate the elements that are processed:

```
  MENU  HELP
┌──────────────────────────────────────────────────────────────────────┐
│                          Command Progress Display                    │
│                                                                      │
│ ............... Fragment Z996AAPG built for APG CICS_SHARED          │
│ ............... Fragment Z995AAPG built for APG CICS1                │
│ ............... Fragment Z994AAPG built for APG CICS1TG              │
│ ............... Application Group CMD_RECEIVER has no automation name│
│ ..........cont. specified.                                           │
│ ............... Fragment Z992AAPG built for APG CPSM                 │
│ ............... Fragment Z991AAPG built for APG DB21                 │
│ ............... Fragment Z99NAAPG built for APG DM22                 │
│ ............... Fragment Z990AAPG built for APG D911                 4│
│ ............... Fragment Z990AAPG built for APG FTP_DAEMON_GROUP     │
│                                                                      │
│                                                                      │
│ PF 1=HELP     2=SPLIT    3=END      4=RETURN   5=RFIND    6=RCHANGE  │
│ PF 7=UP       8=DOWN     9=SWAP    10=LEFT    11=RIGHT               │
└──────────────────────────────────────────────────────────────────────┘
/AOFBUILD JOB
```

On completion, you get a `Build Successful` message back. You can use option 4 to review the Build Report (which is all of the messages that scrolled past while the build was occurring).

Distributing your policy

The SOCNTL data set that you are building the policy into holds automation policy for ALL of the systems that are defined in the policy database.

However, the SOCNTL data set resides on the system you used to create and build the policy. You need to allocate production SOCNTL data sets on the systems where you are going to deploy SA z/OS. Make arrangements to copy the data over to them. You can use TSO XMIT and manually receive them, or use FTP or similar options.

See Chapter 6, “Operations ,” on page 75 for some considerations about how to manage the SOCNTL data on the deployed system.

Maintenance

You need to update your automation policy over time, and as a result of testing.

You can copy your entire PDB data set to make a backup.

- Use the Customization Dialog to change your policy.
- Rebuild and redistribute your policy when you are done.

See Chapter 6, “Operations ,” on page 75 on managing the updates on the target systems to avoid incidents.

Chapter 6. Operations

SA z/OS uses a Manager/Agent model to automate applications on computer systems. It performs primary start or stop and recover automation for applications, and also performs operating system automation. The Automation Agent has a Processor Operations (ProcOps) focal point function that provides automation for Central Processor Complexes (CPCs), related Logical Partitions (LPARs), and certain operating system operations, especially at times when no system operation is active, for example during the nucleus initialization program (NIP) phase on course of Initial Program Load.

SA z/OS works as follows:

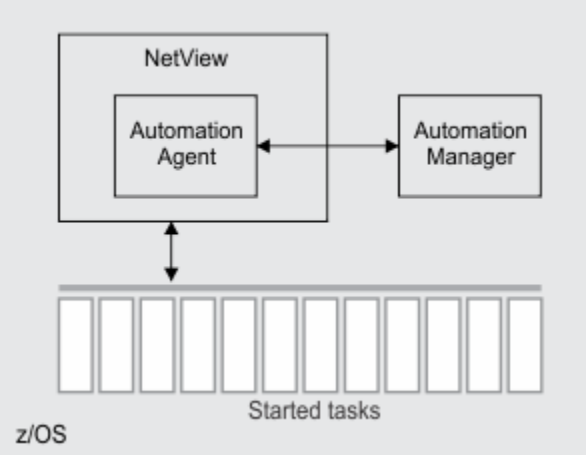

Figure 12. Manager/Agent Overview in SA z/OS

1. Messages get generated from Applications and the Operating System.
2. The messages get routed into NetView from where they are handled by the Automation Agent.
3. The Automation Agent then determines the nature of the message and the response appropriate to the message.
4. Local automation messages trigger an automated response back to the z/OS system from within the agent itself.
5. State change messages trigger an update to the Automation Manager.
6. After the Automation Manager processed the state change and worked out its implications, it sends some orders down to the Automation Agent.
7. The Automation Agent receives the orders and runs them, possibly resulting in commands that are being issued to the z/OS system.

The user interface for managing applications is a NetView console. However, you must use the operating system consoles to actually stop and restart the Automation Manager and NetView with the Automation Agent.

Console operations

These operations are required to get SA z/OS operational.

They are performed at an MVS Master Console, accessible either directly or through System Display and Search Facility (SDSF). You can automate these operations within the z/OS IPL sequence with a COMMNDxx PARMLIB member after you are satisfied with your SA z/OS installation.

The Automation Manager's started task

Naming conventions

The name of your Automation Manager started task is confirmed by your system programmer after the Configuration Assistant is used. The Configuration Assistant generates the INGEAMSA start procedure by default, which is used in the following examples.

Starting the Automation Manager

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

```
S INGEAMSA,JOBNAME=AM,SUB=MSTR
```

More parameters are passed at run time. They are documented in "Starting and Stopping SA z/OS" in *IBM Z System Automation User's Guide*. Your system programmer sets up the task so that no additional parameters are required.

Although the SUB=MSTR parameter is not strictly required, you must use this parameter if you are going to have SA z/OS automate the start, recovery, and shutdown of JES and other subsystems that run the SUB=MSTR parameter. You can omit it for testing, although you must conduct a test with it applied, as the SUB=MSTR parameter is a more restrictive environment for JCL.

Starting a secondary automation manager

The Automation Manager is designed to run in a HOT backup configuration. The first started task becomes the primary manager and runs automation. A second instance of the task is then started to provide a failover capability if the primary Automation Manager fails.

Your system programmer advises you whether this step is required. Typically it would be handled by SA z/OS itself, but under some circumstances it might be manually required. A typical start command for it would be as follows:

```
S INGEAMSA,JOBNAME=AM2,SUB=MSTR
```

The command runs a second instance of the Automation Manager under a different job name.

Again, the SUB=MSTR parameter can be omitted for testing.

Stopping the Automation Manager

To stop the Automation Manager to recycle it, use command: `P AM`.

To stop the Automation Manager when you cannot wait for work in progress to be completed, use:

`F AM,STOP,IMMED` or `F AM,STOP,FORCE`

If you are shutting down the system, SA z/OS is able to shut itself down, but if it cannot shut down, use command: `F AM,STOP,DEFER`.

This command instructs the Automation Manager to keep running until the agent connected to it has shut down and then the Automation Manager stops itself.

During production use, you usually do not have to stop the Automation Manager. It should send itself a **STOP,DEFER** before it shuts down the NetView component. During testing, you might have to shut it down manually.

Additional operations

There are other commands that the Automation Manager responds to. They are documented in "Starting and Stopping SA z/OS" in *IBM Z System Automation User's Guide*.

The Automation Agent's started task

Naming conventions

The Configuration Assistant generates a start procedure named INGENVSA by default. This procedure is used in the following examples.

The agent task requires a subsystem interface task to be started before the agent task is started. The name of the subsystem interface task is also provided by your system programmer. The Configuration Assistant generates the CNMSJ010 procedure by default. This procedure is used in the following examples.

Starting the Automation Agent

The SA z/OS Automation Agent is started with a standard MVS start command. Your system programmer sets it up, so only minimal extra parameters are required.

If SA z/OS is being used to automate the system (as opposed to testing), it must be started with the SUB=MSTR parameter.

Before you can start the agent task, start its subsystem interface (SSI) task first. To do so, issue this command:

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

After the task is initialized, the following message appears:

```
CNM541I NETVIEW SUBSYSTEM ssi_name IS FULLY FUNCTIONAL
```

You can start the Agent task with this command:

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

Failure to specify the SUB=MSTR parameter results in the command being unable to properly automate JES, VLF, LLA, and anything else running the SUB=MSTR parameter. You may omit the parameter for testing however. Ensure that you do a test with the SUB=MSTR parameter as the parameter creates a more restrictive environment for JCL.

Stopping the Automation Agent

You can issue stop commands from either the NetView console or from a z/OS console by replying to the SA z/OS Agent's outstanding WTOR:

```
*0004 DSI802A CNM01 REPLY WITH VALID NCCF SYSTEM OPERATOR COMMAND
```

Usually, you use the following command from a NetView console to stop the SA z/OS Agent:

```
CLOSE STOP
```

The instruction causes the NetView component to shut down all operator logins, autotasks, and other started tasks. The instruction likely causes your NetView login to terminate when it is issued.

If the SA z/OS Agent does not shut down, you can issue the following stronger shutdown command by replying to the NetView outstanding WTOR in the following way, where *nn* is the reply identifier for the WTOR message DSI802A or DSI803A:

```
R nn,CLOSE IMMED
```

The following instruction causes the NetView component to abend any tasks conditionally that are not terminated. If the SA z/OS Agent still is not shut down, try the command:

```
R nn,CLOSE ABEND
```

The following instruction causes the main NetView address space to abend and terminate. The subsystem address space does not require termination. If you must stop it to clean up after testing, use the following stop command:

```
P SYSVSSI
```

If this instruction does not stop the subsystem address space, then leave it running. This address space does not interfere with anything and attempting to stop it more aggressively can cause you to have to restart the system.

Operating SA z/OS

After the Automation Manager and automation agent are started and initialized, you can log on to the agent to operate SA z/OS.

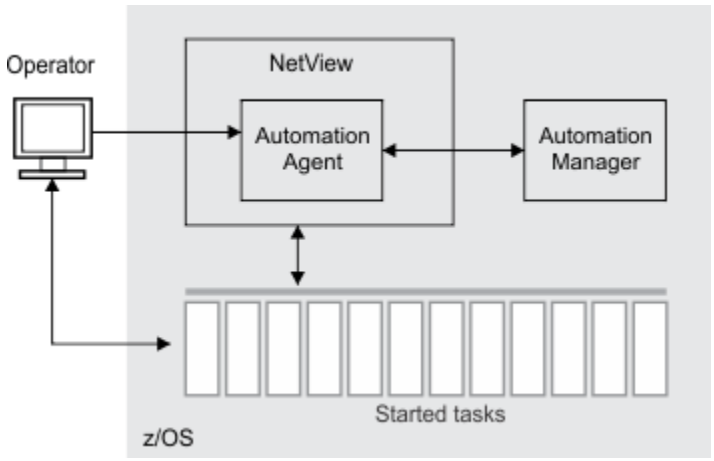

Figure 13. z/OS task structure and the Automation Agent

The SA z/OS agent hosts its user interface environment. Figure 13 on page 78 shows how an operator can communicate to both the Automation Manager and the started tasks.

NetView console

The SA z/OS Automation Agent is built under a NetView environment. To use the agent to interact with automation, you need to log on to the NetView console.

Log on to the NetView console

Your system programmer advises you how to reach the NetView login panel. The login panel looks 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 = IPSNM                     NV63 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
```

Enter your user ID and password into the fields and press Enter. Eventually, you are on the NetView console similar to this one:

```
AOFPOPER                SA z/OS - System Operations
Domain Id . : IPUFJ   ------------ MAIN MENU -----------   Date . . : 09/17/18
Operator Id : YUE             SYSTEM = AOC4                Time . . : 09:16:48

       Select               Description                   Component

         1             Operator Interface                    OPER
         2             Command Dialogs                       CD
         3             Status Display Facility               SDF
         I             IMS Automation                        IMS
         C             CICS Automation                       CICS
         T             TWS Automation                        TWS
         L             User defined Local Functions

     SA z/OS Version 4 Release 2
     Licensed Materials - Property of IBM
     © Copyright IBM Corp. 1990, 2019



Command ===> ____________________________________________________________________
F1=Help     F2=End      F3=Return                                  F6=Roll
                                                                  F12=Retrieve
```

Logging off

To log off, enter LOGOFF in the **Input** field and press Enter. After you are logged out, log back in.

Help

- To access the assistance for a command or an explanation of a message, enter the following command:

  HELP *subject*, where *subject* represents a command name or a message ID.

  A full screen help panel appears. Press PF7 and PF8 to scroll and PF3 to return to the NetView console.
- To get a list of the commands that you can use in the SA z/OS environment, enter this command:

  HELP COMMANDS

  On the panel that is displayed, place the cursor on any row and press Enter to display a detailed description in full screen.
- To display all the help that is associated with the NetView console, issue the following command:

  INDEX

ROLL key

The NetView console provides multiple interactive commands that run immediately. However, the console shows you the 3270 interface for a single command. To switch between different commands, press the ROLL key, usually PF6.

When you start a new program, previous versions of the program stop running, replacing the older version in the roll stack. If you issue the DISPINFO command, then issue the INGRELS command followed by the DISPINFO command again and press PF3 twice will return you to the NCCF console. The first instance of the DISPINFO command is automatically stopped when you issue the command a second time.

Infrastructure

There are two commands that you can use to review the infrastructure that SA z/OS uses to perform its automation. The commands that you use are INGAMS and DISPSYS.

INGAMS command

The INGAMS command displays the agent and the manager, indicates whether they have their automation policy loaded, shows any backup managers that are active, and also refreshes the configuration data used by the Automation Manager.

On the NetView console, enter the **INGAMS** command. You then see the following panel:

```
 INGKYAM0                  SA z/OS - Command Dialogs        Line 1     of 3
 Domain Id . : IPUFJ   ------------ INGAMS   ------------   Date . . : 09/04/19
 Operator Id : OPER1           Sysplex = AOC4PLEX           Time . . : 05:05:17

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

 CMD System   Member    Role  Status     Sysplex  XCF Group Release Comm E2E
 --- -------- --------- ----- ---------- -------- --------- ------- ---- ---
     AOC4     AOC4$$$$3 SAM   READY      AOC4PLEX INGXSGA4  V4R2M0  XCF
     AOC4     AOC4$$$$2 PAM   READY      AOC4PLEX INGXSGA4  V4R2M0  XCF  YES
     AOC4     AOC4      AGENT READY      AOC4PLEX INGXSGA4  V4R2M0  XCF


Command ===> ____________________________________________________________________
F1=Help    F2=End     F3=Return                                  F6=Roll
                      F9=Refresh   F10=Previous   F11=Next     F12=Retrieve
```

What is important is that there is a Primary Automation Manager (PAM) and an AGENT and both have a READY status. The Secondary Automation Manager (SAM) is a backup manager.

You can exit the display by pressing PF3.

DISPSYS command

The DISPSYS command displays information about the system that the command is issued on.

On the NetView console, enter the **DISPSYS** command. You see the following panel:

```
AOFKADAE                  SA z/OS - Command Dialogs        Line 1     of 59
Domain Id . : IPUFJ   ------------ DISPSYS  ------------   Date . . : 12/04/19
Operator Id : YUE             System  = AOC4               Time . . : 05:18:26

 System           : AOC4           in Sysplex  : AOC4PLEX
 Domain           : IPUFJ
 Sysplex Group    : AOC4PLEX
 XCF Group  name  : INGXSGA4
 XCF PlexID       : INGPX$A4

 Software
    Operating System   : z/OS 02.02.00
    NetView            : IBM Z NetView V6R3
       Tower(s)        : SA TCPIPCOLLECT DVIPA
    Automation         : IBM Z System Automation V4R2M0
       Tower(s)        : SYSOPS PROCOPS EMUBMC EMUCA

 Configuration
    Data set     : BUMU.E2E.V420.ACF(ACFZ999)
    Built by     : BUMU 08/27/19 13:19:41
    Activated    : 12/04/19 00:00:31
    CFG Token    : 20190827131928FF097FD72827

 Message automation table(s)
     DSITBL01 SWTCHTAB INGMSGSA INGMSG01

 Flags
    Automation   : Yes
    Init Start   : Yes
    Start        : Yes
    Recovery     : Yes
    Terminate    : Yes
    Restart      : Yes

 Scheduling Subsystem    : JES2
    Type                 : JES2

 Root for SDF updates    : AOC4
 SDF actual  focal point : IPUFJ
     primary focal point : IPUFJ
     backup  focal point : IPUFK

 USS path        : /local/SA/420/daily/usr/lpp/ing/ussauto/lib/
 Inform list     : SDF USR

 IPL Complete Considerations
   Time Limit    : 24:00:00
   Expected Statuses :
                  AVAILABLE
   Important Resources :
                  - None -

 Captured Messages  (maximum is 20)
   2019-09-03 11:14:49 :
      ING377I The E2E agent service INGRE2E failed with RC=229,
      REASON=CGlobals AOF_E2EAGT* corrupted

 WLM Capacity Data
    Time stamp of WLM query      : 2019-12-04 05:09
    Total SUs last 10 minutes    : 126789331
    Used SUs last 10 minutes     : 155753
    Resources with DS=AVAILABLE  : 68
    Expected free SUs            : 126618515

Command ===> ____________________________________________________________
F1=Help     F2=End      F3=Return                          F6=Roll
F7=Backward F8=Forward  F9=Refresh                        F12=Retrieve
```

The most important information on the panel is the identification of the automation policy that is being used. The identification is the **Configuration Data set** in the middle of the panel. This information is also available on the **INGAMS Details** panel (line command B).

To scroll the pane, use PF7 and PF8. To exit the panel, press PF3.

Resources

Resources are the elements that SA z/OS uses to model started tasks and other elements of the z/OS system that SA z/OS can automate. The Automation Manager interface works in terms of resources.

All resources have 2- or 3-part hierarchical names:

- `name/type`
- `name/type/system`

The resource names are derived from the entities created by the Customization Dialog and how these entities are linked with systems, groups, or application groups.

The type identifies the nature of the resource. Table 5 on page 82 shows the most frequently used resource types. For a full list of the resource types, see "Entry Type Introduction" in *IBM Z System Automation Defining Automation Policy*.

Table 5. Resource types (partial list)

Resource Code	Resource Type	Resource Description
APL	Application	A started task or other automateable entity, typically a started task
APG	Application Group	A group of resources that are defined and managed within the Automation Manager
MTR	Monitor	A resource that represents a monitor
SYS	System	A resource that represents a z/OS system
GRP	Group	A group of one or more systems

The system qualifier is present for all resources that are linked to a specific system.

There are four commands to view data about these resources:

- INGLIST and INGINFO, which show data from the Automation Manager perspective
- DISPSTAT and DISPINFO, which show data from the Automation Agent perspective

There are also two commands that can be used to dynamically observe the resources. See “Monitoring resources” on page 91.

- SDF, which shows a dynamic dashboard
- INGNTFY, which sends messages to your NetView console every time a resource changes status

INGLIST command

The INGLIST command queries the resources that are known to the Automation Manager to view some information about them and to initiate actions against them.

To start the **INGLIST** interface, type the following command:

```
INGLIST resource
```

where *resource* is an optional parameter that represents the name of the resource that you want to inquire about. It can include wildcards, for example, `INGLIST RMF*`.

The command produces a response such as this:

```
INGKYST0                  SA z/OS - Command Dialogs        Line 1     of 4
Domain Id . : IPUFJ   ------------ INGLIST  ------------   Date . . : 09/05/18
Operator Id : YUE             Sysplex = AOC4PLEX           Time . . : 08:44:08
 A Update   B Start    C Stop     D INGRELS  E INGVOTE  F INGINFO  G Members
 H DISPTRG  I INGSCHED J INGGROUP K INGCICS  L INGIMS   M DISPMTR  P INGPAC
 R Resume   S Suspend  T INGTWS   U User     X INGWHY   / scroll
CMD Name        Type System   Sus Compound     Desired      Observed   Nature
--- ----------- ---- -------- --- ------------ ------------ ---------- ------
___ RMF         APL  AOC4         SATISFACTORY AVAILABLE    AVAILABLE
___ RMF         APL  AOC5         SATISFACTORY AVAILABLE    AVAILABLE
___ RMFGAT      APL  AOC4         SATISFACTORY AVAILABLE    AVAILABLE
___ RMFGAT      APL  AOC5         SATISFACTORY AVAILABLE    AVAILABLE




Command ===> ____________________________________________________________
F1=Help     F2=End      F3=Return     F4=DISPSTAT   F5=Filters    F6=Roll
                        F9=Refresh   F10=Previous  F11=Next      F12=Retrieve
```

You can enter commands against individual resources in the **CMD** column on the left side of the panel. The primary commands that you probably want to use are B (Start), C (Stop), E (INGVOTE) , F (INGINFO), and X (INGWHY).

INGINFO command

You can either start the INGINFO command from the **INGLIST** panel or you can start it from the command line: `INGINFO resource`.

The *resource* is the name of the resource that you want to know about. The INGINFO command works with a single resource at a time. If the resource name you specify resolves to multiple resources, make a selection. For example, issue the following command:

```
INGINFO RMF*
```

The results might be as follows:

```
AOFKSEL4                  SA z/OS - Command Dialogs        Line 1     of 4
Domain Id . : IPUFJ   ------------ INGINFO  ------------   Date . . : 09/05/18
Operator Id : YUE                                          Time . . : 09:24:16

Multiple instances found for RMF*
Select one item to be processed, then press ENTER.

Sel.  Name         Type System    Description
----  -----------  ---- --------  ----------------------------------------
___   RMF          APL  AOC4      Resource Measurement Facility
___   RMF          APL  AOC5      Resource Measurement Facility
___   RMFGAT       APL  AOC4      RMF Monitor III Data Gatherer
___   RMFGAT       APL  AOC5      RMF Monitor III Data Gatherer
```

Select one of the resources by typing an S under the **SEL** column and pressing Enter.

The **INGINFO** panel shows lots of detailed information about a resource:

```
INGKYIN0                  SA z/OS - Command Dialogs        Line 1     of 171
Domain Id . : IPUFJ   ------------ INGINFO  ------------   Date . . : 08/24/18
Operator Id : YUE             Sysplex = AOC4PLEX           Time . . : 09:26:49

 Resource    =>  RMF/APL/AOC4_____________________   format: name/type/system
 Target      =>  ________    System name, domain ID or sysplex name

 Resource      : RMF/APL/AOC4
 Description   : Resource Measurement
Facility

  Status...
     Observed          : AVAILABLE
     Desired           : AVAILABLE
     Automation        : IDLE
     Startability      : YES
     Compound          : SATISFACTORY     Last changed : 2018-08-23 01:44:13
     Health Status     : N/A

  Dependencies...
     PreStart          : Satisfied
     Start             : Satisfied
     PreStop           : Satisfied
     Stop              : Unsatisfied
     Startability      : Satisfied

  Trigger              : -None-

  Inherited Group Trigger(s)...
```

You can scroll up and down the panel by pressing PF7 and PF8. To exit this panel, press PF3.

INGINFO display

The INGINFO command output displays the following sections:

Statuses

The Status block gives you status information about the resource:

```
     Observed          : AVAILABLE
     Desired           : AVAILABLE
     Automation        : IDLE
     Startability      : YES
     Compound          : SATISFACTORY
     Health Status     : N/A
```

The statuses are as follows:

Table 6. Resource Status Overview

Status	**Description**
Observed	Current status of the resource, as reported by the Automation Agent.
Desired	The goal state for the resource. It is up (AVAILABLE) or down (UNAVAILABLE).
Automation	What the Automation Manager recognizes the agent is doing for the resource at the moment. Usually it is IDLE, which means that the agent is not doing anything as it receives no orders.
Startability	An estimation by the Automation Manager of whether it is possible to start the resource. If not YES, then there is a problem with one or more prerequisite resources that make it impossible to start this resource.
Compound	An aggregate status, synthesized from all of the other statuses to give an overall opinion on the state of the resource. SATISFACTORY is good, but only means that the resource is in its goal state, not necessarily that it is up.

Table 6. Resource Status Overview (continued)	
Status	**Description**
Health Status	A status that is calculated from the health status that is returned by health monitor resources. N/A means that there is no health monitor resource that is attached to the resource.

For a full description of these statuses, see "Statuses Supplied by the Automation Manager" in the *User's Guide*.

Resource dependencies

The Dependency block gives you information about the state of the resource's prerequisite resources:

```
Dependencies...
   PreStart          : Satisfied
   Start             : Satisfied
   PreStop           : Satisfied
   Stop              : Unsatisfied
   Startability      : Satisfied
```

The conditions to perform each action, `PreStart`, `Start`, `PreStop`, and `Stop`, are evaluated over all resources that are defined as being related to this resource in your automation policy. The panel shows the result of that evaluation.

If a dependency is `Satisfied`, then the corresponding action can be taken, when necessary. If it is unsatisfied, then it cannot be taken. In this example, the Stop dependency is unsatisfied. The Automation Manager does not inform the Agent to issue the commands to stop the resource until the stop dependency is satisfied. The stop dependency becoming satisfied is not sufficient, on its own, to cause the Automation Manager to tell the agent to stop the resource. There are a number of other elements that affect that decision, primary among which is the resource's desired state.

A prerequisite resource is a resource that has to be in a certain state before something specific can be done to this resource. Most resources are independent of each other. However, where there is a dependency between the resources, a relationship is present in the automation model to represent this dependency.

Some typical relationships would be as follows:

```
MakeAvailable/WhenAvailable
PrepUnavailable/WhenAssumedDown
```

The first relationship indicates that the supporting resource must be up before this dependent resource can be started. The second relationship indicates that you cannot issue the commands to prepare the dependent resource to be shut down until the supporting resource is down or can be safely assumed to be down. For more information about relationship, see "Relationships" on page 5.

History

The History section is at the bottom of the **INGINFO** panel and looks like as follows:

```
History       :
   2018-09-05 00:00:42  System Registration Delay
     HSAL6347I Orders are not suspended
     HSAL6113I Resource may send orders
     HSAL6334I Agent registered
     HSAL6335I Agent registration timer purged
   2018-09-05 00:00:30  Automation Agent is suspended
     HSAL6346I Orders are suspended
     HSAL6333I Starting agent registration timer
     HSAL6114I Resource may not send orders
     HSAL6172I Group Observer update sent
   2018-09-04 00:00:45  System Registration Delay
     HSAL6347I Orders are not suspended
     HSAL6113I Resource may send orders
     HSAL6334I Agent registered
     HSAL6335I Agent registration timer purged
```

Each major entry represents a work item that the Automation Manager processed. That is, an update from the Automation Agent. Between the work items are HSAL messages that are generated from the automation logic engine.

The messages record:

- Observations, such as a change in a resources status, for example : `HSAL6260I Status/Observed is Starting`.
- Conclusions that are reached, for example, `HSAL6315I Shutdown supporters are active`.
- Decisions that are made: `HSAL6126I Resource can be started`.
- Actions that are taken, for example: `HSAL6152I Make Available order sent`.

The history sequence that is illustrated shows messages that are displayed towards the end of an automated startup of RMF. The bold lines show what is happening on the Automation Agent with the oldest ones at the bottom. The messages in between are sorted with the oldest ones at the top of each group.

If things are not working properly, the history sequence shows valuable information for problem determination.

DISPSTAT command

The agent also has its own, internal representation of the resources that are being automated. The name of the agent's resource is the same as the name of the manager's resource, but without the type and system qualifiers. Agent resources are known, for historical reasons, as subsystems.

For example, the agent resource for `RMF/APL/KEY1` is `RMF`.

Only APL resources have corresponding agent subsystems.

The DISPSTAT command displays a summary of the agent's information about one or more SUBSYSTEMs. You can start DISPSTAT from the command line. You can also press PF4 in the INGLIST interface to transfer to a DISPSTAT view for the same set of resources.

The command's parameter on the command line is a list of one or more SUBSYSTEMS, which can include wildcards:

```
DISPSTAT RMF*
```

Pressing Enter then produces a panel such as this:

```
 INGKYDS0                  SA z/OS - Command Dialogs        Line 1     of 2
 Domain Id . : IPUFJ   ------------ DISPSTAT ------------   Date . . : 09/05/18
 Operator Id : YUE             System  = AOC4               Time . . : 10:29:24
  A dispflgs  B setstate  C ingreq-stop  D thresholds  E explain  F info G tree
  H trigger   I service  J all children  K children  L all parents  M parents
 CMD Resource    Status   System   Jobname  A I S R T RS Type     Activity
 --- ----------- -------- -------- -------- - - - - - -- -------- ---------
  _  RMF         UP       AOC4     RMF      - - - - - -  MVS      --none--
  _  RMFGAT      UP       AOC4     RMFGAT   - - - - - -  MVS      --none--




 Command
===> ____________________________________________________________________

 F1=Help     F2=End      F3=Return     F4=INGLIST    F5=Filters    F6=Roll
                         F9=Refresh   F10=Previous  F11=Next      F12=Retrieve
```

The important information here is the resource (SUBSYSTEM) name, status, MVS job name, and the Activity field. Pressing PF10 and PF11 scrolls through a couple of extra information panels and PF7 and PF8 scrolls the list up and down.

The **CMD** field on the left lets you enter commands against a resource. The most important are SETSTATE (B: reset the state of the resource), EXPLAIN (E: display detailed information about the resources current state) and INFO (F: display detailed information about the resource).

The J, K, L, and M commands display children and parent information based on the relationships that are defined between the resources. For example, if you entered M against RMF, then JES2 would appear in the panel.

DISPINFO command

The DISPINFO command is the agent's detailed resource information panel.

You can initiate it from the command line, from inside the **DISPSTAT** panel (line command F) or by pressing PF4 in the **INGINFO** panel. If you do the latter, you are shown the agent resource corresponding to the AM resource that you were looking at.

From the command line, enter the following command:

```
DISPINFO RMF
```

This command takes only a single resource. So if the string that you specified resolves to more than one resource, you are presented with a selection panel, where you can refine your resource specification.

The panel that is displayed looks as follows:

```
AOFKINFO                SA z/OS  - Command Dialogs      Line  1    of 118
 Domain ID   = IPUFJ     -------- DISPINFO ----------     Date = 08/21/18
 Operator ID = YUE                                       Time =
06:14:42

  Subsystem  =>  RMF         Target  =>  AOC4     System name, domain ID
                                                 or sysplex name
  Subsystem   : RMF         on System : AOC4

  Description : Resource Measurement Facility

  Inform list : SDF USR

  Class chain : C_APPL       Class for general APL definitions

  Job Name    : RMF
  ASID        : 01C1

  Job Type    : MVS

  Current status     : UP
    Last Monitored   : 10:25:54 on 09/05/18
    Last Changed     : 11:14:48 on 09/03/18
    Last Message
      AOF571I 11:14:48 : RMF SUBSYSTEM STATUS FOR JOB RMF IS UP -
              CHANGED FROM DOWN AT SYS-OPS
REFRESH

  Monitor            : INGPJMON
    Monitor Status   : ACTIVE
    Monitor Interval : None specified


 Command ===> ____________________________________________________________
 F1=Help     F2=End      F3=Return     F4=INGINFO                 F6=Roll
             F8=Forward  F9=Refresh                              F12=Retrieve
```

DISPINFO display

The DISPINFO command displays the following information:

Current status

The Current status section shows the resource's current agent status, when it was last updated, and the message that is associated with that status change.

There are many agent statuses, but the primary flows that are listed are:

- A normal flow for a resource that is being started: `AUTODOWN -> RESTART -> STARTED -> ACTIVE -> UP`
- A normal flow for a resource that is being stopped: `UP -> AUTOTERM -> AUTODOWN`

Table 7 on page 88 shows the statuses indicating the automated recovery is impending.

Table 7. Agent status that indicates automated recovery is impending

Status	Description
STOPPING	The resource is being shut down outside of automation. Depending on your automation policy settings, SA z/OS might try to restart it.
ABENDING	The resource failed and is shutting down. Again, SA z/OS might try to restart it.
INACTIVE	When SA z/OS ran a monitor to check the resource automation status and detected that it wasn't running, the monitor reported that the resource was not active. SA z/OS tries to restart it.

Table 8 on page 89 shows the statuses indicating there is a problem that automation cannot fix on its own. The operators should examine the resource and get it to a state from which it can be restarted. They might have to call in support.

Table 8. Statuses indicating manual intervention is required

Status	Description
STARTED2	The resource is started, but it did not issue the message indicating that it is up yet. If the resource is just running slow, verify it is really active and manually set its status to UP using SETSTATE.
HALTED	The resource is still running, but it has encountered a problem that impacts its functionality. This might fix itself.
BREAKING	The resource failed and is shut down. The nature of the failure indicates that SA z/OS might not automatically restart it.
BROKEN	Similar to BREAKING, but the resource is down.
ZOMBIE	The resource's final termination message is received by SA z/OS, but the address space is not terminated within a reasonable time frame.
STUCK	SA z/OS issues all the shutdown commands that it has for the resource, but it has not shut down.
STOPPED	The resource is shut down outside of automation and SA z/OS is instructed not to automatically restart it.
CTLDOWN	The resource is down and SA z/OS is instructed not to start it. The agent does not treat this situation as an error.
DENIED	The agent automation flag for the resource is turned off, preventing SA z/OS from acting for the resource. You can use the INGAUTO or INGSUSPD command to re-enable automation for the resource after you determine that it is appropriate to do so.

Flags

Flags show the current, effective setting of the Automation Agent flags for this resource.

```
Flags - Automation : Yes
        Init Start : Yes
        Start      : Yes
        Recovery   : Yes
        Terminate  : Yes
        Restart    : Yes
```

Each flag controls a different aspect of automation (Start After IPL, Start from Down, Recovery, Shutdown, Restart). If the AUTOMATION flag is turned off, no automation is allowed, regardless of how the other flags are set. You can use the INGSUSPD command to suspend the automation of a resource completely. On the Automation Agent side, the AUTOMATION flag is effectively set off. Manipulating the AUTOMATION flag using INGAUTO or DISPFLGS is possible for compatibility reasons, but not recommended due to possible side-effects.

Thresholds

There are some thresholds that are defined in the automation policy for the resource.

```
   Thresholds   -   SUBSYSTEM
   Critical   : 02 in 00:05:00
   Frequent   : 02 in 00:30:00
   Infrequent : 02 in 01:00:00
```

These definitions are counted in terms of application failures (unsolicited shutdowns or recoverable abends) within the time specified. If the Critical threshold for a resource is exceeded (in this case, if the abend occurs more than twice within 5 minutes), then the automated recovery is suspended and the resource's state is changed to BROKEN to await operator assessment.

Startup and Shutdown Details

This section gives the details of how SA z/OS is instructed to start and stop the resource. In addition to checking that the instructions are correct, it can also act as a reference if you want to perform a startup or a shutdown manually. Use INGSUSPD to suspend all automation for the resource during this process.

```
Active Startup : None

Active Shutdown : None

Start Up Mechanism  -
  Scheduling Subsystem : JES2

  External startup     : NEVER

  Start Up Process -
    Prestart :
      None

    Startup :

      Start command to be used  :
        MVS S RMF

      User Start Up Commands :
        None specified

   Poststart :
      None

   Refreshstart :
      None

   Anystart :
      None

  Start Delay          : 00:02:00
  Start Cycles         : 1

Shutdown Mechanism -

  External shutdown    : NEVER
```

Captured Messages

The Captured Messages section shows the last few messages that the agent is processing for the resource:

```
Captured Messages for RMF  (maximum is 20)  -

  2018-09-03 11:14:48 :
     AOF571I 11:14:48 : RMF SUBSYSTEM STATUS FOR JOB RMF IS UP -
             CHANGED FROM DOWN AT SYS-OPS REFRESH
```

The colors of the messages indicate their severity - Green (Normal), Yellow (Unusual), and Red (Critical).

If the resource ends with an abend, you can use this section to see the captured messages. Use the timestamp to find correlated messages in the Syslog and Netlog.

Policy definitions

The Policy definition section shows automated actions that SA z/OS is instructed to perform beyond a simple automated restart.

This example tells SA z/OS that when the ERB603D WTOR is issued during a normal RMF start, it automatically responds GO to it.

```
Policy Definitions for RMF -

  ERB306D :
    REPLY=(NORM,,'GO')
```

These actions are mostly controlled by the Recovery automation flag. If you do not want SA z/OS to act, use the INGSUSPD command to suspend the resource.

Monitoring resources

Although you can monitor resources by repeatedly refreshing and reorganizing the INGLIST panel, SA z/OS provides some mechanisms to do monitoring more efficiently.

Status Display Facility (SDF)

Status Display Facility (SDF) is a dynamically updated dashboard view of your system. Your automation administrator can change the layout of the dashboard information. What you see could be different from the examples that are shown here.

You typically use SDF to monitor the system during normal operations to look for problems. If you have multiple systems, there is a focal pointing mechanism that can concentrate all of the dynamic updates to a single dashboard. Here, you can monitor all of the systems.

To access SDF, type the **SDF** command at a NetView command line. The high-level resources panel might look as Figure 14 on page 91. Cool colors (BLUE, GREEN) are used for normal statuses. More neutral colors (CYAN, WHITE) are used for intermediate statuses. Hot colors (YELLOW, PINK, RED) are used for critical statuses.

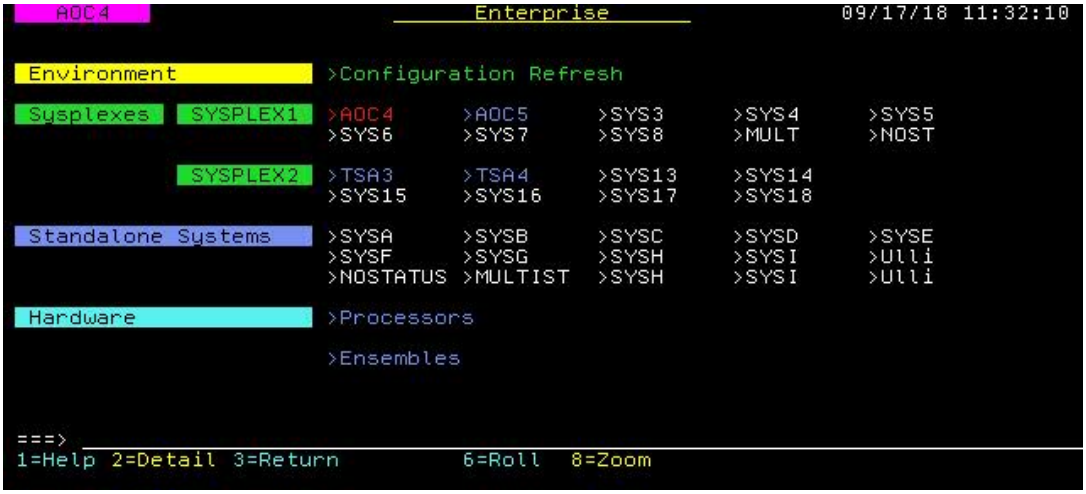

Figure 14. Example SDF top-level panel

Detail status panel (reached by PF2)

In Figure 14 on page 91, AOC4 is highlighted in red, which means that a resource on AOC4 is BROKEN. To view more information about the broken resource on this system, select AOC4 by moving the cursor to it and pressing PF2 to display a Detail Status Display panel. It gives a detailed description of the broken resource with the highest status priority, for example, Figure 15 on page 92. The detail records are maintained as an ordered stack and you may scroll up and down this stack with the PF7 and PF8 keys. Press PF3 to return to the dynamic SDF panels.

Figure 15. Example status display

Resource status panel (reached by PF8)

On the top-level SDF panel, pressing PF8 when you select a system shows a Resource Summary Status panel that gives an overview of the different resources, messages, and components that are being monitored. Figure 16 on page 92 shows the resource statuses of AOC4 in this example.

Figure 16. Example summary status

The dynamic fields, prefixed with a greater than symbol (>) have a color according to the worst status contained within them. To view further details about APPLS in this example, move cursor over it and press PF8 to show each individual component status, as shown in Figure 17 on page 92.

Figure 17. Example subsystem status

To go back up to the higher level panel, press PF7. If the list of resources is too large to fit on the screen, press PF10 and PF11 to scroll through the list.

The PF13 through PF24 keys can be set to issue SA z/OS commands directly against a resource. Put the cursor over the resource and press the Shift plus the PF Key.

To see more details about an item, place the cursor over the item and press PF2. A detail record is displayed for the status change event.

INGNTFY command

The INGNTFY command turns on and off message notification. When the command is turned on, the messages that you select to see are sent to your NetView console. INGNTFY is set in the policy database through the Customization Dialog.

The INGNTFY command provides the following functions:

Autowrap

The autowrap function controls whether your terminal is held when the screen is full of data, or if the screen is automatically overlaid with new data. If you do not have autowrap turned on, you need to press enter to clear information on the panel.

To turn on autowrap, issue `AUTOWRAP 10`.

This instruction makes the NetView console wait 10 seconds before the screen is cleared of information and shows the next page of messages that are buffered. You can set other time if you want. Setting a value of 0 flushes the buffer and returns to real-time monitoring.

To stop autowrap, issue `AUTOWRAP NO`.

This instruction means that you have to manually press Enter to clear the information on the screen and see the next set of messages.

Getting messages

To monitor messages, you can enter the INGNTFY command. A list of all currently defined Notify Operators are displayed. Some operators might be predefined in your automation policy.

```
AOFKAANT                  SA z/OS  - Command Dialogs      Line  1    of 1
Domain ID   = IPUFJ     ---------- INGNTFY  ----------    Date = 09/16/18
Operator ID = YUE                                        Time = 08:25:00

Cmd:  A Add    C Show/Change settings    D Delete    O Turn OFF msg receiving

Cmd Operator System   Log Rcv Description          Classes
--- -------- -------- --- --- -------------------- ----------------------------
 _  SYSOPS   AOC4      Y   Y  MVS CONSOLE          44 45 46 50 51 52 80




Command ===> ____________________________________________________________________
F1=Help     F2=End      F3=Return                                  F6=Roll
                        F9=Refresh                                F12=Retrieve
```

Messages that are monitored by the SYSOPS pseudo-id get echoed onto the SysLog and Master Consoles.

To add yourself as a new message receiver, enter an A as a command against any of the existing entries.

```
AOFKAASN                  SA z/OS  - Command Dialogs
Domain ID   = IPUFJ     ---------- INGNTFY  ----------    Date = 09/16/18
Operator ID = YUE                                        Time = 08:28:16

 Status/Action =>  ADD___      Operator status:
                                 ON     - Set your notifications ON
                                 OFF    - Set your notifications OFF
                                 ADD    - Add or Modify an operator
                                 DELETE - Remove an operator
                                 QUERY  - Look up an operator
 Operator ID   =>  ______   Operator for ADD, DELETE or QUERY
 Classes       =>  44 45 46 50 51 52 80 ____________________________________
 Description   =>  MVS CONSOLE ____________

 Held Messages - Information       => _    Immediate Action   => _
                 Eventual Action   => _    Immediate Decision => _
                 System Wait       => _

 Target        =>  AOC4____    System name, domain ID or sysplex name

AOF710A VERIFY/REVISE INPUT AND THEN PRESS ENTER
Command ===> ____________________________________________________________________
F1=Help     F2=End      F3=Return                                  F6=Roll
                                                                   F12=Retrieve
```

Messages within SA z/OS are organized into classes, and they are used on the **INGNTFY** panel to select the messages that you want to see.

Table 9. SA z/OS message classes (partial)

Class code	Description
40	All Automation Agent messages. OK for testing, but can become congested when automation is working hard.
46	User intervention is required. Automation wants an operator to do something.
50	Critical Threshold exceeded messages. Relevant resources are automatically recovered.
80	All Automation Manager messages. OK for testing, but can become congested when automation is working hard.
81	Intervention is required. Issued for resources where SA z/OS thinks operator intervention is required.
82	Automation is denied. SA z/OS wants to do something to a resource but the automation flag is off.
90	Infrastructure. Issued if the Aumation Manager encounters a problem.

A basic set of messages for an operator to monitor are 46, 50, 81, 82 and 90. For a full table of the message classes, see "Message Classes" in *IBM Z System Automation Messages and Codes*.

On the INGNTFY panel, specify your NetView user ID, the classes of messages you want to receive, and then press Enter.

Press PF3 to exit the panel and press PF9 to refresh the list of notification operators.

Stopping and restarting the message flow

If you want to stop receiving messages, use the o command code on the **INGNTFY** panel.

From the NetView console, you can type this command to turn off messages: `INGNTFY OFF`.

You can turn back on messages by typing this command: `INGNTFY ON`.

There is no confirmation message when used this way.

Message display

If you are a messages receiver defined in the INGNTFY and message notification is enabled, the monitored messages that are displayed on your NetView console appear as follows:

```
NV63 SA42 NM             Tivoli NetView   IPUFJ YUE      09/18/19 08:39:37
| IPUFJ    HSAL6012E SYSVAPPL/APL/AOC5; INTERVENTION REQUIRED; AUTOMATION
           INHIBITED
| IPUFJ    HSAL6012E SYSVIPLC/APL/AOC5; INTERVENTION REQUIRED; AUTOMATION
           INHIBITED
| IPUFJ    HSAL6012E SYSVSSI/APL/AOC5; INTERVENTION REQUIRED; AUTOMATION
           INHIBITED
| IPUFJ    HSAL6012E TCPIP/APL/AOC5; INTERVENTION REQUIRED; AUTOMATION
           INHIBITED
| IPUFJ    HSAL6012E TSO/APL/AOC5; INTERVENTION REQUIRED; AUTOMATION INHIBITED
| IPUFJ    HSAL6012E VLF/APL/AOC5; INTERVENTION REQUIRED; AUTOMATION INHIBITED
| IPUFJ    HSAL6012E VTAM/APL/AOC5; INTERVENTION REQUIRED; AUTOMATION INHIBITE
| IPUFJ    HSAL6012E JES2MON/MTR/AOC5; INTERVENTION REQUIRED; AUTOMATION
           INHIBITED
| IPUFJ    HSAL6012E AOCAPLEX/DMN; INTERVENTION REQUIRED; AUTOMATION INHIBITED
| IPUFJ    HSAL6012E LNXDMN/DMN; INTERVENTION REQUIRED; AUTOMATION INHIBITED
| IPUFJ    HSAL6012E CSM_AOCA/REF/AOCAPLEX; INTERVENTION REQUIRED; AUTOMATION
           INHIBITED
| IPUFJ    HSAL6012E CSM_AOCB/REF/AOCAPLEX; INTERVENTION REQUIRED; AUTOMATION
           INHIBITED
=X= ***
```

The messages are color-coded according to severity – Green (OK), Yellow (Warning), Red (Problem).

Held messages

If you asked for the messages to be held on the **INGNTFY** panel, the messages move to the top of the panel when the screen is cleared. They remain there until you place your cursor on them and press Enter.

You can use the DM utility command to remove the messages: `DM *`.

This command deletes all held messages. In general, you hold important messages.

Message help

You can use help to get details about a message: HELP *messageID*. For example, **HELP HSAL6020I** returns the corresponding message explanation.

```
HSAL6020I resource; AWAITING AUTOMATION

Explanation:  The resource is automatable, but is waiting for
prerequisites to be satisfied.  These could be either dependencies or
triggers.

    The variable resource shows the name of the resource.

System Action:  Automation waits until the prerequisites for automation
are satisfied and then starts processing.

Operator Response:  If the prerequisites involve operator action, for
example, manually setting a trigger, check that this was done.

System Programmer Response:  None.

Classes: 80, 84.
```

Starting and stopping resources

Default availability

In your automation policy, each resource has a Default Desired Status setting.

The possible values for the setting are as follows:

Table 10. Default Desired Status settings	
Default Desired Status setting	**Description**
Always	The resource is always up. In the absence of any other instruction, the resource is started and is recovered if it fails. This is usually the most common setting.
On Demand	The resource is up if it is needed and it is down otherwise. If the resource is needed to support a resource that you must start, then this resource is started and recovered if it fails. If the demand for it goes away, it is shut down.
As Is	Automation does not start or stop the resource unless instructed to do so. If the resource is running, it is left alone, but it is not recovered if it fails. If it is down, it is also left alone. Use this status setting for resources that you do not want automation to take action.

There is some inference of these settings through the parent-child tree, with the Automation Manager working out an effective Default Desired State for each resource. The INGINFO panel shows the value:

```
      Desired Available    : Always
```

The Default Desired Status provides the background availability goals that you are working with as an operator. It explains that the automation with the on demand setting can take your commands and *amplify* them, if, for example, you ask SA z/OS to stop the only resource that is providing demand for a chain of on demand resources. Then SA z/OS shuts down all the resources in the chain in response to your request to stop the one resource at the bottom. SA z/OS is working to minimize the dependencies in running your system. If it causes operational problems, then discuss the matter with your automation administrator.

Groups

Groups of resources are covered in “Application Groups” on page 100, but note that any non-passive group overrides the Default Desired Status settings of all of its members. You start and stop groups the same way that you start and stop application resources.

Persistent requests

The mechanism that SA z/OS provides for operators to request resources to be started and stopped is a persistent request-based mechanism.

When you request a resource to be started or stopped, SA z/OS stores that fact. The requests interface with the automation by changing the resource’s Desired Status setting. If there is a request from an operator against the resource, its Desired Status setting is set to the value that the operator requested.

If you ask for a resource to be stopped, SA z/OS stops it. It is kept down until you cancel that request. If you ask for a resource to be started, SA z/OS brings it up and recovers it if it fails. It is kept up until you cancel that request. When you tell SA z/OS to stop or start a resource, you are changing its Desired state (overriding the default Desired state) until further notice.

The expectation here is that all regular starting and stopping of resources is automated through one or more SA z/OS schedules, the default settings, or both. So operators take direct control of a resource to stop it or start it under exceptional circumstances. The correct action after the operators complete their action is to cancel the request against the resource to return control of it to SA z/OS. The operators must *not* issue a request for the counter action. Use request and cancel, not request and counter-request.

Request priorities

If there are multiple requests against a resource from different sources, for example, an operator and a schedule, then the request priority is used to determine which request wins and actually sets the resources Desired status. The priorities are divided into three broad bands, LOW, HIGH and FORCE. FORCE is only used by operators and for certain scenarios where an automated shutdown of a dependent resource is required in response to the failure of the supporting resource. Within each priority band, requests from the Operator have the highest priority.

For example, if you put a DOWN request in at 6:00 a.m. (causing the resource to be stopped), and at 7:00 a.m. a schedule puts in an UP request. Then, the resource remains down and is not started at 7:00 a.m. because the higher priority operator request is still present against the resource.

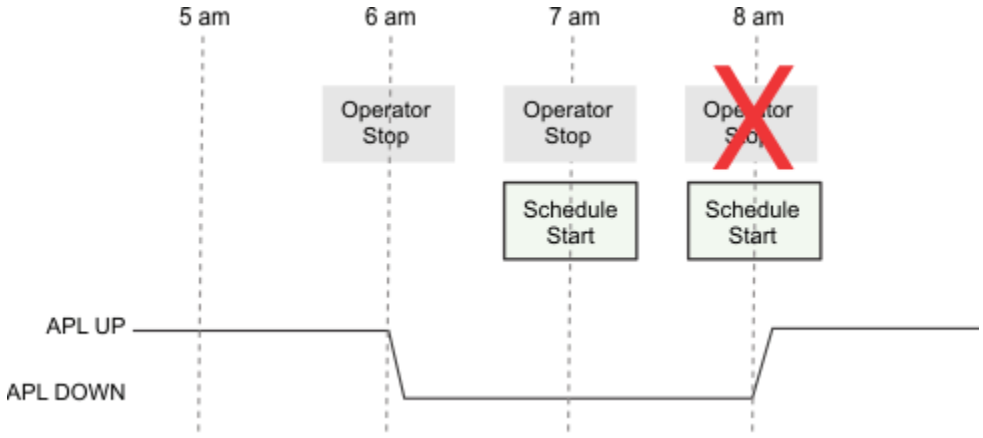

Figure 18. Operator Request Priorities to hold down Resource

When you complete the necessary actions to bring the resource down, cancel your stop request to return the resource to SA z/OS's control. SA z/OS can then start it in response to, for example, a schedule request.

Votes

A request asks for the Desired state of a resource to be set to a specific state. The automation model, however, contains many resources that are connected by various dependencies, which generally serve to prevent resources from being started and stopped. Consider a request to stop JES2; it has many children, all of which place a condition on JES2 shutdown that the children are shut down first. Expecting an operator to know and manually request that each of these children is stopped is unreasonable. Thus, objects that represent the request and its action get propagated through the resource structure. They are called *votes*.

The propagation of votes to resources depends on the action that the votes represent and the relationships that are present within the structure. Votes for stop requests tend to be propagated to child resources that are dependent on the stopping of the resource. Votes for start requests tend to get propagated to parents that are supporters of the start of the resource.

Each resource examines all of the votes that are passed to it and selects the one with the highest priority. The priority comes from the request that originated the vote. In a tie breaker, for example, votes from two or more requests from the same source (OPERATOR, SCHEDULE, and so on) are present, and a vote to be UP beats a vote to be DOWN.

Locked votes

When you are working with SA z/OS, you might encounter a *locked vote*. Locked votes occur because the resources did not reach a finite state yet. When you request SA z/OS to stop a resource, it starts issuing commands from a list of shutdown commands until the resource actually comes down. If the resource does not stop, SA z/OS posts it to an error status. While the agent is working, the vote that triggered the request against the resource is locked. No further change can occur. The agent goes from

half way through shutting a resource down to starting it. At the end of the shutdown process (successful or otherwise), the vote is unlocked. You then replace it with a more forceful shutdown sequence if it fails.

Starts follow a similar process, but during a start you can override the lock by specifying INTERRUPT=YES on your INGREQ request, replacing the start vote with, for example, a shutdown vote. However, there is no guarantee that the default shutdown sequence actually works.

INGREQ command

INGREQ is the main dialog for entering requests to be sent to the Automation Manager.

You can enter a request directly, through any of the available front ends or from the **INGLIST** panel with commands B (Start) or C (Stop). When not started directly, some of the input fields are prefilled.

```
INGKYRU0                  SA z/OS  - Command Dialogs      Page   1 of 2
Domain ID   = IPUFJ     ---------- INGREQ   ----------    Date = 09/07/18
Operator ID = YUE                                        Time = 10:03:50

  Resource    =>  RMF/APL/AOC4_____________________  format: name/type/system
  Target      =>  ________    System name, domain ID or sysplex name

  Request     =>  ______      Request type (START/UP, STOP/DOWN or CANCEL)
  Type        =>  NORM______  Type of processing (NORM/IMMED/FORCE/user) or ?
  Scope       =>  _________ Request scope (ONLY/CHILDREN/ALL)
  Priority    =>  LOW____     Priority of request (FORCE/HIGH/LOW)
  Expire      =>  __________ , _______ Expiration date(yyyy-mm-dd), time(hh:mm)
  AutoRemove  =>  ____________________________________  - see help for list
  Restart     =>  NO__        Restart resource after shutdown (YES/NO)
  Override    =>  NO_____________________  (ALL/NO/TRG/FLG/DPY/STS)
  Verify      =>  YES_        Check affected resources (YES/NO/WTOR)
  Comment     =>  ____________________________________________________________
                  ____________________________________________________________
  Appl Parms  =>  ____________________________________________________________

AOF710A VERIFY/REVISE INPUT AND THEN PRESS ENTER
Command ===> ____________________________________________________________
F1=Help     F2=End      F3=Return                          F6=Roll
            F8=NextPnl                                    F12=Retrieve
```

The most important things on the panel are the resource name, the requested action (START or STOP) and the Restart option.

To start a resource, enter its resource name, START as the request and press Enter. To stop a resource, do the same but with STOP as the request.

You now see a verification panel:

```
 AOFKVFY1                  SA z/OS - Command Dialogs        Line 1     of 2
 Domain Id . : IPUFJ   ------------ INGREQ   ------------   Date . . : 09/07/18
 Operator Id : YUE                                         Time . . :
10:28:19

 Verify affected resources for request STOP

 CMD: S show overrides   T show trigger details   V show votes
 Cmd Name        Type System   TRG SVP  W  Action Type       Observed Stat
 --- ----------- ---- -------- --- ---- -  ------ ---------- -------------
  _  RMF          APL  AOC4             Y  STOP   NORM       AVAILABLE
  _  RMFGAT       APL  AOC4             Y  STOP   NORM       AVAILABLE




 Command ===> ____________________________________________________________
 F1=Help     F2=End      F3=Return                          F6=Roll
                       F9=Refresh   F10=GO        F11=CANCEL    F12=Retrieve
```

The panel lists the resources that are affected by the action, the resources that votes are directly propagated to. A stop vote is propagated to the children resource RMFGAT/APL/AOC4. Yes (Y) under **W** column indicates that the vote is considered by SA z/OS as the "winning" vote, the highest priority vote on the resource.

Verify whether the list of affected resources is acceptable or not. If acceptable, press PF10 to send the request to the Automation Manager. It then takes the actions that you requested.

```
 AOFKMSG0                 SA z/OS - Command Dialogs        Line 1     of 2
Domain Id . : IPUFJ   ------------ INGREQ -------------   Date . . : 09/07/18
Operator Id : YUE                                        Time . . : 10:45:12

Sel System   Message
--- -------- -------------------------------------------------------------------
    AOC4     AOF302I    07:45:12 : REQUEST INGREQ STOP BY YUE IS COMPLETED FOR
                        RMF/APL/AOC4
```

Next, a message is displayed indicating your request is completed.

INGVOTE command – purging your request

The request that you just added with the INGREQ dialog remains in the Automation Manager until you remove it. To remove the vote, use the INGVOTE command.

The INGVOTE command runs against a single resource and shows you all of the votes and requests that accumulated against it.

```
 INGKYRQ0                  SA z/OS  - Command Dialogs       Line  1    of 5
 Domain Id . : IPUFJ   ------------ INGVOTE  ------------   Date . . : 09/07/18
 Operator Id : YUE             Sysplex = AOC4PLEX           Time . . : 10:45:26

  Resource    =>  RMF/APL/AOC4_____________________
  Target      =>  ____________   System name, domain ID or sysplex name

  Desired Available..: Always

 Cmd:  C cancel request   K  Kill request    S show request details
 Cmd Action WIN Request/Vote Data
 --- ------ --- ------------------------------------------------------------
  _  STOP    Y  Request    : MakeUnAvailable
                Created    : 2018-09-07  10:50:13
                Originator : OPER_YUE(YUE)
                Priority   : 01720000     Should Be Down - Operator
                Status     : Winning/
Satisfied



 Command ===> ____________________________________________________________

 F1=Help     F2=End      F3=Return                              F6=Roll
                         F9=Refresh                             F12=Retrieve
```

In this case, the request is entered against RMF, so you can kill it directly by entering a K against it. (C does the same thing, but goes through a verification panel).

If you look at RMFGAT before canceling the request, you see this information:

```
INGKYRQ0                  SA z/OS  - Command Dialogs       Line  1    of 5
Domain Id . : IPUFJ   ------------ INGVOTE  ------------   Date . . : 09/07/18
Operator Id : YUE             Sysplex = AOC4PLEX           Time . . : 12:37:30

 Resource    =>  RMFGAT/APL/AOC4_____________________
 Target      =>  ____________   System name, domain ID or sysplex name

 Desired Available..: Always

Cmd:  C cancel request   K  Kill request    S show request details
Cmd Action WIN Request/Vote Data
--- ------ --- ------------------------------------------------------------
    STOP    Y  Vote       : MakeUnAvailable
               From Req.  : MakeUnAvailable for RMF/APL/AOC4
               Created    : 2018-09-07  10:50:13
               Originator : OPERATOR_YUE(YUE)
               Priority   : 01720000     Should Be Down - Operator



Command ===> ____________________________________________________________

F1=Help     F2=End      F3=Return                              F6=Roll
                        F9=Refresh                             F12=Retrieve
```

The INGVOTE command indicates that it is a vote, not a request, and an input field is not displayed for it. The **From Req.** field indicates the resource that the request was entered against. You have to run the INGREQ REQ=CANCEL command against that resource if you want to cancel the request behind the vote.

Application Groups

Application Groups (APG) are active automation elements that manage the availability of a small set of applications.

For example, a group might be all the resources that are needed to run a CICS or an IMS. Or, at a higher level, a group might be a complete application, consisting of subgroups to manage the complex resources that are needed to make the application work.

The advantages of groups are that they automatically manage the availability of the resources they are looking after. They provide you with a single point of control to request that the resources are stopped or started. The down side is that if an operator does not tell the groups what he wants, the groups can end up working at cross purposes.

Application group types

There are three types of Application Groups (APG) available within SA z/OS:

BASIC
: In a BASIC application group, all of its components must be available before the group is considered to be available.

MOVE
: In a MOVE application group, exactly one of its components must be available before the group is considered to be available. In MOVE groups, you can specify alternative components to start if their primary component fails to start.

SERVER
: In a SERVER application group, any number of components can be specified that must be available before the group is considered to be available. In SERVER groups, you can specify what happens if their components fail to start.

For more information of application group, see “Grouping support” on page 5.

Relationships

Relationships for a group work on the group and are passed onto the group’s members.

If a group is a child of a resource, every member of the group is also considered to be a child of the resource, when SA z/OS works out what can be stopped and started. There is an exception for Move and Server groups. Move and Server groups allow their members to be stopped (provided the group as a whole is not being stopped) while their stop dependency is unsatisfied. The set of active members of the Move and Server groups can be changed.

Runmodes

Runmodes are not groups, but they can provide a similar function at some levels.

Runmodes are defined as sets of runtokens that are used to tag individual resources. You then select the runmode that you want to be active on a system. The INGRUN command can be used to activate or deactivate a particular runmode. It is suggested that runmodes rather than large groups of unrelated resources are used for "bulk" application control on a system.

For more information about Runmodes, see "Using Runmodes" in *IBM Z System Automation User's Guide*.

Operating Application Groups

You start and stop groups with the same commands, INGREQ and INGVOTE, that you use to operate standard APL resources. These commands request that the group is started or stopped according to its predefined policies.

Operating Group Members

Operations on group members are not recommended because your instructions are in conflict with the groups.

Move and Server groups activate a different resource, while Basic groups decide to shut down the whole group. All types of groups can override your directive. Or, the groups delay a directive until some recovery action is taken to reduce its impact upon availability.

If you must use application groups, you can use the INGREQ and INGVOTE commands as for any other resource; however, watch for unexpected reactions to your instructions.

Managing Application Groups

With Basic groups, there is nothing that you can do to isolate individual members. With Move and Server groups, you have the option of modifying the groups preferences to temporarily override the policy set for the group in the Customization Dialog.

You can use this facility to, for example, change the member that a move group selects to be active.

To do so, issue the INGGROUP command and specify the name of the group that you want to manage.

```
INGKYGRA                  SA z/OS  - Command Dialogs
Domain ID   = IPUFH     ---------- INGGROUP ----------    Date = 09/08/18
Operator ID = YUE             Sysplex = AOC2PLEX          Time = 09:01:30

Specify or revise the following data:

 Target    => _________  System name, domain ID or sysplex name

 Action    => ___________ EXCLUDE-AVOID-INCLUDE or ACTIVATE-PACIFY or
                          ADJUST or RESET DEFAULT or OVERRIDES, POLICY or
                          RECYCLE-CANCEL

 Opt.Parms => ____________________________________________________________

 Group(s)  => PNEST/APG___________________________________________________
              ____________________________________________________________
              ____________________________________________________________
              ____________________________________________________________
 System(s) => ____________________________________________________________
              ____________________________________________________________
Command ===> ____________________________________________________________
F1=Help     F2=End      F3=Return     F4=Members                 F6=Roll
                                     F10=GO                    F12=Retrieve
```

Press PF4 to show the members and the policy for them:

```
 INGKYGRB                  SA z/OS - Command Dialogs        Line 1     of 2
 Domain Id . : IPUFH   ------------ INGGROUP ------------   Date . . : 09/18/18
 Operator Id : YUE             Sysplex = AOC2PLEX           Time . . : 09:24:40

  Group: PNEST/APG                Nature: Server  Passive: NO    Suspend:
  Description: SERVER APG, AVT2, members NEST1, NEST2
  Excluded :
  Avoided  :
  Mode     : Normal    Availability Target:  2    Adjust: 0     Result=> 2
                       Satisfactory Target=>      Adjust: 0
  Rolling Recycle: None

  Name         Type System    Pref  Adj    Result  Avl   Eff   Stat  Act  Sus
  -----------  ---- --------  ----  -----  ------  ----  ----  ----  ---  ---
  PNEST1       APG  AOC2       700      0     700   Yes   950   Sel   --   --
  PNEST2       APG  AOC2       700      0     700   Yes   950   Sel   --   --









 Command ===> ____________________________________________________________
 F1=Help     F2=End      F3=Return                                 F6=Roll
                         F9=Refresh   F10=GO        F11=Reset     F12=Retrieve
```

The **Result** column can be edited. If you enter a new preference value for the member, the software calculates an adjustment to be applied to the base policy value to make the resulting preference the value that you specified. The **Act** column indicates any actions that are likely to be taken as a result of applying your change.

- To force a member to be stopped, enter a result of 1.
- To force a member to be started, enter a result greater than the highest value in the **Eff** column.
- To return to the default values specified in your automation policy, press PF11.

A table of valid preference values is documented in the *IBM Z System Automation User's Guide*.

After the changes, press PF10 to activate those changes. If you exit the dialog with PF3, your changes are discarded.

You can also, for a Server group, change the values of the Availability Target and Satisfactory Target fields. The Availability target sets the number of group members that are started. The Satisfactory target specifies a lower number that have to be started for the group's dependencies to be considered satisfied. For example, if five members are started, you can set the satisfactory target to 2 to begin starting resources that are dependent on the group while the remaining three members are still initializing.

These panels are also available for Basic groups, but you cannot change anything of Basic groups.

Automation policy management

Customization and policies

To work, SA z/OS requires a model of your application workload and some policy definitions that characterize your environment.

This information is entered through an ISPF application that is called the Customization Dialog that is provided as a part of SA z/OS. The data in the model is stored in a data set known as a Policy Database (PDB). The PDB gets built (compiled) into another data set called System Operations Configuration File (SOCNTL) that is used by SA z/OS at run time. The SOCNTL file then must be copied onto each system that loads its automation policy from it.

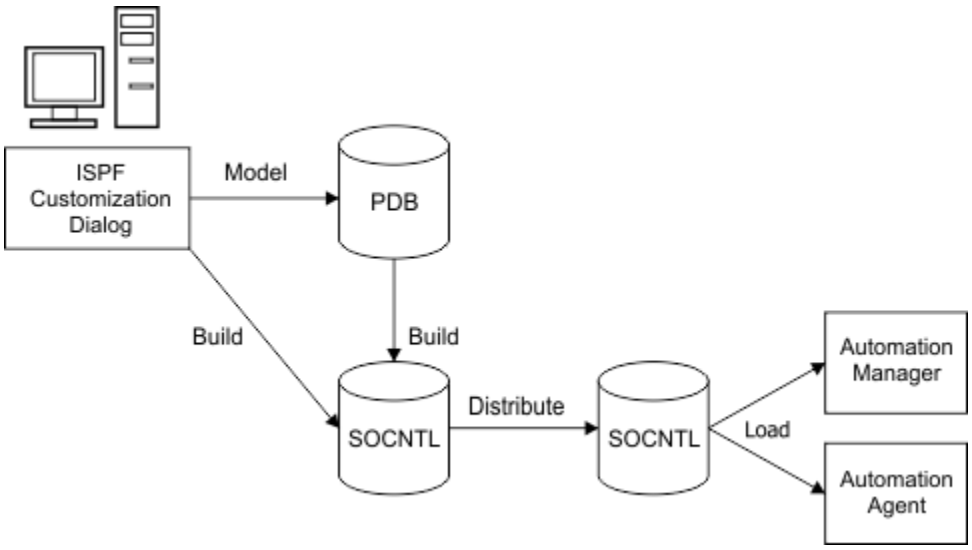

Figure 19. Automation Policy Lifecycle

The SOCNTL file is defined to the Automation Manager task, usually through its HSAPRMxx member, which resides in a data set in your systems PARMLIB concatenation. The SOCNTL file can be anywhere on your system where both the Automation Manager and the Automation Agent can read it. Although, you can use a generation data group (GDG) for the deployed version of your SOCNTL file, it is not strictly required.

When the Automation Manager initializes, it pulls the name of the SOCNTL file that the Automation Manager is meant to use out of the HSAPRMxx member and loads. Then, the Automation Manager processes the parts of the SOCNTL file it is supposed to load. An SOCNTL file data set contains data for many systems. So the Automation Manager first determines which files are required for use by both the Automation Manager and the Automation Agent before the Automation Manager loads them.

When the Automation Agent initializes, it requests from the Automation Manager which SOCNTL data set it is meant to use. The Automation Manager responds with the name of the data set that is loaded and the agent then loads its corresponding files from the SOCNTL data set.

Note: The data that is loaded by the Automation Manager includes a Customization Timestamp. The time stamp identifies the last time that the data was edited in the Customization Dialog before the SOCNTL data set was built. The time stamp is included in the response sent to the agent and is checked against the time stamp that the agent extracts from the data that is going to load. If the two time stamps do not match, the agent refuses to proceed with the load and you do not have automation for your system. If this mismatch happens, use the MVS MODIFY command to modify the Automation Manager and request it to load a new automation configuration file.

For example, if your Automation Manager loads an automation policy that you built yesterday, when it comes up and you overwrite the SOCNTL file with a policy that was built today, when you stop and restart the Automation Agent, there is an issue. The Agent refuses to load the automation policy from the SOCNTL file because the time stamps do not match.

To fix the situation, instruct the Automation Manager to load the new configuration data with a modify command:

```
F INGEAMSA,REFRESH,SOCNTL_dataset_name
```

The Automation Manager loads the new configuration data and token and then instructs the agent to load them. Because the tokens now match, the agent can proceed.

COLD, WARM, and HOT starts

The Automation Manager maintains data in the following data sets that are used if it fails or is restarted.

Table 11. Files maintained by the Automation Manager

File	Description
Takeover file	This data set stores a snapshot of the current internal state data within the Automation Manager.
Schedule override file	This data set stores data about service period overrides.
Configuration file	This data set stores the name of the current SOCNTL data set in use.

There are three different start options for the Automation Manager:

Table 12. Actions for different Automation Manager start types

Actions	Description
COLD	Reads the SOCNTL data set name from HSAPRMxx, and erases the Schedule and Takeover data.
WARM	Reads the SOCNTL data set name from the Configuration data set, erases the Takeover data, and restores the schedule data.
HOT	Reads the SOCNTL data set name from the Configuration data set, and restores the Takeover data and the Schedule data.

Following a HOT start, the Automation Agents are instructed to analyze their systems and return status data for all of the applications that the agents are automating.

Whether you start an automation manager from the console using the HOT start type or whether a Secondary Automation Manager (SAM) takes over from the current Primary Automation Manager (PAM), is technically the same. In both cases, the Takeover and Schedule data is used to initialize the automation manager.

If you lose both, the Primary manager and the Secondary manager, you can start a new Automation Manager with a HOT start. The HOT start attempts to take over as a PAM from the data that is left in the files. The command that is used would be as follows:

```
S INGEAMSA,JOBNAME=AM,SUB=MSTR,START=HOT
```

The command does not work if there is already an Automation Manager that is running on the system (because that is already the Primary manager). In this situation, the newly started manager becomes a SAM.

Although it is common practice among customers to hardcode the Automation Manager's start type as HOT, unexpected and unwanted behavior follows a prolonged system or automation outage. The automation gets restored to exactly the state that it was in when the failure occurred. If the Automation Manager was stopping something at the time, it continues to do so after the HOT start is complete. The goal of a HOT start is to enable a SAM to take over if the PAM fails or is shut down. A gap of no more than 60-120 seconds is expected. If the system is restarted between when automation last ran and when it is being started, SA z/OS automatically purges all requests that are flagged as being purged during the system's next IPL.

Manage your System Operations Control File (SOCNTL)

The Automation Manager tracks the location of the SOCNTL data set in its files, and if an agent gets recycled, it asks the Automation Manager the location of the SOCNTL data set.

The Automation Manager then sends it the name of the SOCNTL data set and the customization token it saved from its SOCNTL data. Then, the agent finds the SOCNTL data set on disk and checks its customization token. If you have, for example, overwritten the SOCNTL data set with a newer one, the agent finds it, but does not load it because the customization tokens do not match.

To recover from a token mismatch, use the command that is given at the end of "Customization and policies" on page 103.

If you do not want the new configuration that is loaded now, you can find a copy of the old configuration. If the data set name is different, use the modify command to tell the Automation Manager to reload it. If you cannot do that, you cannot use automation until it is acceptable to activate the new configuration. Alternatively, your automation administrator can attempt to rebuild the old configuration in the Customization Dialog.

As this situation is not preferable, there are two available techniques to manage the versions of the SOCNTL data set.

Simple data sets

If you are going to manage your SOCNTL data set as a simple data set, you need to use a strong naming convention:

Table 13. Outline of an SOCNTL Naming Scheme

Data Set	Description
MY.SOCNTL.OLD	The data set you last used in production.
MY.SOCNTL.CURRENT	The data set you are currently using in production.
MY.SOCNTL.NEW	The next data set to use.

New SOCNTL data sets are installed by overwriting the MY.SOCNTL.NEW data set. When you activate a new configuration, copy CURRENT over OLD and NEW over CURRENT, then load the updated configuration. The copying is best done with a batch job. To instruct the Automation Manager to load the overwritten policy from inside the Automation Agent, use the INGAMS command:

```
INGAMS REFRESH,CFG=*
```

INGAMS instructs the Automation Manager to reload its configuration from the data set that it is using (which is MY.SOCNTL.CURRENT). If you want to switch it to the old SOCNTL, use:

```
INGAMS REFRESH,CFG='MY.SOCNTL.OLD'
```

Remember to change the setting back to MY.SOCNTL.CURRENT when it is appropriate to do so.

Generation Data Groups

It is recommended that you use a Generation Data Group (GDG) to hold your SOCNTL data. The GDG avoids many of the complications of using simple data sets. Although working with them is a little more complicated, you generally have to use batch jobs to update them. The benefit significantly outweighs the cost.

A GDG is a data set that automatically maintains its history. When you copy a new SOCNTL data set on top of it, the old one is still kept as an accessible data set. You can configure the number of generations of the data set that are kept. Typically five are used for SOCNTLs, although you can have more if you want.

The command to load an SOCNTL from a GDG is:

```
F AM,REFRESH,MY.SOCNTL.GDG(0)
```

or

```
INGAMS REFRESH,CFG=MY.SOCNTL.GDG(0)
```

The commands instruct it to load the current generation. What it stores to its file, however, is the actual name of the file on disk, MY.SOCNTL.GDG.G0012V00, for example. If an updated SOCNTL is written to the GDG, it is saved as file MY.SOCNTL.GDG.G0013V00. If the agent must reload the old configuration (for example, generation 12) the file is still available on disk for the agent to load it. The 12th data set is deleted after generation 18 is created (assuming you are keeping five generations).

To activate a new configuration with GDGs, issue:

```
INGAMS REFRESH,CFG=*(0)
```

The current generation of the GDG is available.

To step back to an older configuration, issue:

```
INGAMS REFRESH,CFG=*(-1)
```

It is up to you to track which older generation is the one you want to revert to, if the new configuration does not work. The INGAMS 'Show Details' subcommand (B) shows you which data set you are currently using when issued against the PAM:

```
INGKYAM3                  SA z/OS  - Command Dialogs      Line  23   of 51
Domain ID   = IPUFJ     ---------- INGAMS   ----------    Date = 09/14/19
Operator ID = YUE             Sysplex = AOC4PLEX          Time = 05:27:12

 Diagnostic Info
   Snapshot size          : 4868036
   Number of resources    : 229
   Number of requests     : 37
   Number group requests  : 34
   History records        : 192987
   Max History records    : 262144

 Config dataset name : BUMU.E2E.V420.ACF
 Config member       : HSAZ997
                           Z997CRES  STRUCTURE   20190914105130
                           Z997CSCH  SCHEDULE    20190914105130
                           Z999CLGC  STRUCTURE   20190914104636
 Config token        : 20190914105130FF097FD72827
 Config version      : 01



Command ===> ____________________________________________________________
F1=Help     F2=End      F3=Return                      F6=Roll
F7=Backward             F9=Refresh                    F12=Retrieve
```

You have to scroll down to see the **Config dataset name**.

The risk when you use GDGs is that you copy so many "new" SOCNTL data sets in, your current SOCNTL data set falls off the bottom of the stack. Use a staging data set as a remedy and copy the SOCNTL data into the GDG just before you go to activate it. If you find your current data set is getting close to the bottom of the stack, you can copy it into a temporary data set. Then, write it back to the top of the stack and tell the Automation Manager to switch to it.

Chapter 7. Processor Hardware Interfaces

With the Processor Hardware Interfaces' System Automation component, users can manage IBM mainframe hardware at an enterprise scope.

By working with this component, you can achieve the following purposes:

- Consolidate, reduce, and automate manual Hardware Management Console (HMC) operations for IBM Z hardware.
- Use the same user interface and automation environments that System Automation provides for application and system management.
- Satisfy specific hardware operation and automation requirements by having the choice between two interfaces, different in function scope and connectivity.

Related Hardware Documentation

The following documentation contains processor hardware information, relevant to the processor hardware interface:

- *IBM Z SNMP Application Programming Interfaces*

Go to the IBM Resource Link (Library tab) online facility to download the publications that are valid for your processor.

Introduction and Overview

The processor hardware interfaces provide the automation and operation base to monitor and control IBM Z mainframe processor hardware with the System Automation product. The two interfaces are listed as follows:

Processor Operations

Processor Operations is based on TCP/IP and SNMP protocols. Processor Operations runs hardware commands and receives hardware events from the supported mainframes. Instead of using TCP/IP and SNMP, communication can also be redirected by using the Base Control Program internal interface (BCPii) which is not dependent on external network protocols.

INTERNAL connection path

As the identifier 'internal' indicates, this connection path uses a connection protocol that is provided by z/OS, called Base Control Program internal interface (BCPii).

Mainframes That Are Supported

IBM Z (IBM z13/z13s or later).

Scope of Control

Enterprise-wide processor scope.

Connectivity and Functionality at a Glance

System Automation can monitor and control all Central Processor Complexes (CPCs). That includes all configured LPARs, systems, and z/VM guests that are running on these CPCs.

Figure 20 on page 108 shows a type 8561 z15 processor.

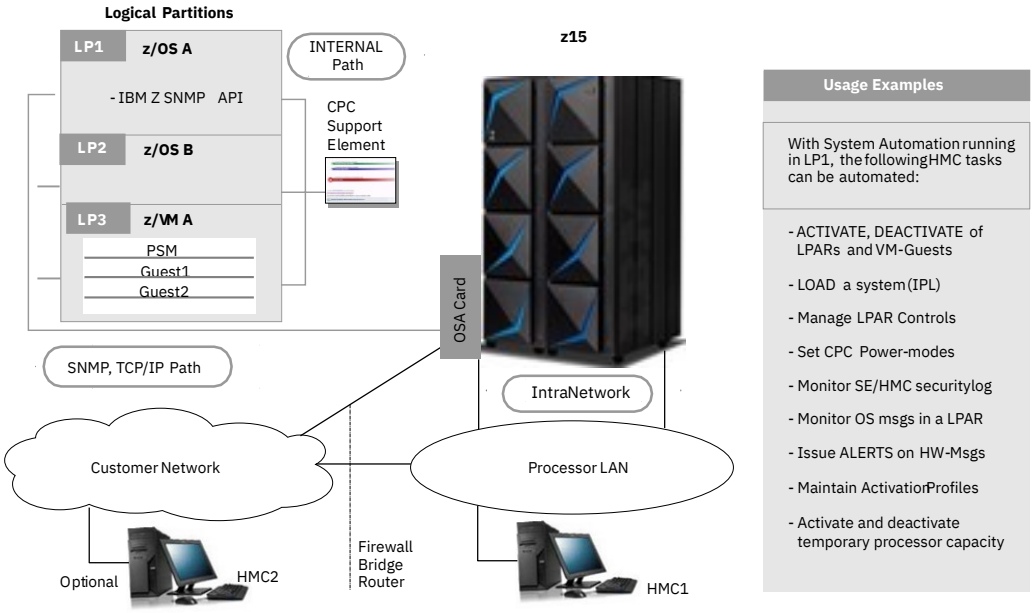

Figure 20. Processor Hardware Interfaces

Figure Annotations

z15
Other supported IBM Z mainframe types (IBM z13/z13s or later) are also valid.

CPC Support Element (SE)
The processor hardware interfaces use the Systems Management functions of the SE to operate the CPC and its LPARs. SE customization is required before the processor hardware interfaces can access and use the SE.

HMC1, HMC2
Hardware Management Consoles. At least one HMC is attached and operational in a valid CPC environment. The HMC must be customized before the processor hardware interfaces can use the console.

IBM Z SNMP API
The processor hardware interfaces of System Automation use this API. See "Related Hardware Documentation" on page 107.

Processor LAN – Customer Network
Together with the CPC and its own Intra Network, a processor LAN environment is available, enabling the SE and HMC communication for one or multiple CPCs. Support Elements are always attached to the processor LAN. An extra HMC, configured for the processor hardware interfaces, might be attached to the Customer Network.

SNMP Path
The Processor Operations hardware interface requires a TCP/IP infrastructure for its SNMP-based communication. It means that in the z/OS LPAR where System Automation is running, an IP Stack must be active with a connection (OSA-card) to the customer network. This enables the communication with an attached HMC. Usually, Firewall/Bridge/Router network components are used

to secure and isolate the two networks. This might require additional customization steps to grant TCP/IP based processor hardware interface communication.

Rather than using TCP/IP and SNMP communication path, you can also define a hostname value of ISQET32. This definition will redirect communication to the CPC and its LPARs over BCPii using INTERNAL path. This is also called 'Hybrid SNMP'.

TCP/IP Path
The communication between Processor Operations and a z/VM guest virtual machine is based on TCP/IP socket service. On the z/VM side, the ProcOps Service Machine (PSM) component of Processor Operations provides this support. See the figure annotation "z/VM, PSM, Guests" on page 109. Similar to the SNMP path, TCP/IP path requires a TCP/IP infrastructure. On the z/VM PSM host side, TCP/IP infrastructure is required.

INTERNAL Path
The Base Control Program internal interface (BCPii) allows communication between the CPC and its LPARs via the SE of the processor. More network elements, such as IP stack, network cards, or routers are not required. For System Automation, the complete physical connection is located inside the CPC cage. Communication to remote CPCs is handled by one or more master HMCs. INTERNAL path is for use by GDPS only.

The BCPii implementation that is used in System Automation is not compatible with the BCPii base component of z/OS. The available z/OS BCPii documentation does not apply for System Automation. Application programs exploiting the z/OS BCPii and System Automation might run concurrently on the same z/OS system.

Logical Partitions
With System Automation active in one LPAR, you can monitor and control each accessible LPAR of your local and remote CPCs. The Processor Operations interface also uses the integrated console of each LPAR for message monitoring and command forwarding. Finally, LPAR hardware commands such as ACTIVATE or LOAD are provided with Processor Operations.

z/OS A, z/OS B, z/VM A
These elements represent the operating systems running in the processor's LPARs. In the Figure 20 on page 108, System Automation is running on partition LP1 on z/OS system A. To monitor and control a system, each one must be defined in the System Automation policy, together with the processor and related connection path (protocol) information.

z/VM, PSM, Guests
The guest machines of a z/VM host, running in an LPAR of the CPC can be managed with Processor Operations. The ProcOps Service Machine (PSM) component of Processor Operations must be installed on the z/VM host system. The PSM provides the required TCP/IP connection services. This support is independent of any z/VM virtual server management that is provided by Unified Resource Manager on IBM zEnterprise Systems.

Usage Examples
Operation, configuration, and recovery tasks for a CPC and its LPARs are performed by using the Hardware Management Console. By using the processor hardware interfaces of System Automation, many of these tasks can be done with the System Automation user interfaces, or run automated.

Preparation and Configuration

Selecting the appropriate Interface to use

To select the appropriate interface to use, you need to consider multiple aspects. One criterion is the interface function set; another is the setup requirements to get the interface working.

Table 14. Required Hardware and Software Setup Tasks

Required Hardware and Software Setup Tasks	**Processor Operations Using TCP/IP and SNMP**	**Processor Operations Using ISQET32**
Microcode level validation	YES (Use IBM Resource Link or contact your IBM Software support to validate if service level recommendations exist for your SE or HMC, related to the API (SNMP, BCPii). If so, the necessary Microcode levels (MCLs) are applied. System Automation has no prerequisite hardware MCLs for its processor hardware interfaces.)	
Customize API settings for Support Element	YES (SE, HMC, or both can be specified as the connection end points for a CPC connection in the System Automation Customization Dialog.)	YES
Customize API settings for Hardware Management Console		NO
Customer network preparation	YES (Regardless of the Processor Operations connection endpoints, SE or HMC, connectivity between the processor LAN and the customer network is always required.)	NO
z/OS IP stack	YES (Processor Operations specifies the IP stack to use for an SE/HMC connection, in case System Automation runs in a multiple IP stack environment. Using a separate IP stack for Processor Operations connections is a way to isolate this IP traffic from regular business traffic.)	NO
Z System Automation customization See Chapter 5, “Customization,” on page 33.	YES	YES

For more information about these tasks, see the HMC console and the *System z Hardware Management Console Operations Guide* of your CPC.

Preparing the Autotasks

The work of the processor hardware interfaces in System Automation is distributed between the individual user operator tasks and sets of specialized automated operator tasks for each interface. System Automation provides all of the required pre-definitions for these autotasks.

For security purposes, these specialized tasks are assigned to the Automated Operator role with INGAUTO as their default group. For more information, see section "Roles" in chapter "Security and Authorization" of *IBM Z System Automation Planning and Installation*.

Processor Operations

The DSIPARM member AOFOPFPO contains the autotask definitions that are used for Processor Operations. There are more autotasks pre-defined than used by System Automation.

Predefined Task Range	Used Tasks
ISQBT001 - ISQBT040	ISQBT001
ISQCM001 - ISQCM040	ISQCM001

Enabling INTERNAL Communication

Usage of the INTERNAL path depends on the activation of the AUTHW001 and AUTHW002 autotasks.

When System Automation is started, the autotasks that are defined as automated function operators in your automation policy entry Automation Operators are activated. It also activates the connection to your local processor and all defined remote processors.

Automated Function Name	Autotask Name	Task Function Description
HWOPER01	AUTHW001	Monitors the Connection
HWOPER02	AUTHW002	Communication task

Create an Automated Operators entry of HW_OPERS and define the HWOPER01 and HWOPER02 automated functions. Select each of the automated functions (HWOPER01, HWOPER02) and type the associated autotask name from the table in the **Primary Operator** field. Leave the other fields empty.

```
               Automation Operator Definitions       Line 00000001 Col 001 075

Entry Type : Automation Operators  PolicyDB Name   : MYPDB
Entry Name : HW_OPERS              Enterprise Name : MYPDB

Line Commands: S (Select), M (Move), B (Before), A (After), R (Repeat)
               I (Insert lines), D (Delete lines)

Cmd AutoFunc Primary Operator  Backup Operator   Messages for this Operator
___ HWOPER01 AUTHW001          ________________  ________________________________
___ HWOPER02 AUTHW002          ________________  ________________________________
```

Finally, the HW_OPERS entry must be assigned to the system, where System Automation is supposed to be active. Use the WHERE USED policy of your HW_OPERS entry to assign the system.

For more policy-related information, see "Entry Type Reference" in *IBM Z System Automation Defining Automation Policy*.

Hardware Preparation

Depending on the interface, use the following documentation references provided.

For Processor Operations, follow the SNMP connection instructions. For INTERNAL path (or hostname ISQET32), follow the Base Control Program internal interface (BCPii) connection instructions.

Always use the highest SE/HMC workplace version information. The information about how to navigate and enter the SE/HMC data is related to its user interface view. System z API customization information can be found in these resources:

- See "Traditional SA z/OS Configuration", section "Preparing the Hardware" in *IBM Z System Automation Planning and Installation*.
- Equivalent virtual hardware preparation of the Processor Operation support for z/VM guest machines, is documented in "Traditional SA z/OS Configuration", section "Preparing the VM PSM" in *IBM Z System Automation Planning and Installation*.

Hardware Resource Security

System Automation provides the operator role concept and the TARGET command parameter-based security options. For the processor hardware and z/VM guest machine resources, an extra authorization layer is implemented. See the appendix "Controlling Access to the Processor Hardware Functions" in *IBM Z System Automation Planning and Installation*.

The definition of a PSM security resource profile is required for the Processor Operations z/VM guest system support. Follow the scheme that is described in the appendix "Controlling Access to the Processor Hardware Functions" of *IBM Z System Automation Planning and Installation*.

Ensure the security network name and security resource name parts of the SAF profile name match what you defined in the System Automation Customization Dialog for this PSM processor type.

Customization

CNMSTYLE Customization

For Processor Operations, make sure that the System Automation TOWER.SA statement in your active CNMSTYLE specification includes keyword PROCOPS. Only if so, Processor Operations runs fully enabled.

See the following example tower definition statement in DSIPARM member CNMSTGEN for the recommended location for your additional configurations:

```
*************************************************************************
* Final SA license specification    *
*************************************************************************
TOWER = SA
TOWER.SA  = SYSOPS license PROCOPS SATELLITE
```

Building Blocks

The building blocks are managed only if the key elements of the processor hardware interfaces are defined in the System Automation policy. Three policy entry types are important for the hardware interfaces: Processor, System, Enterprise.

Processor (PRO)

This entry defines the local and remote processors (CPCs) that you want to control. The entry includes definitions that are required to access the mainframe and details about each processor partition that you want to manage, such as partition name and the system name that is running in this partition.

For example, you define a processor that is called PROC1 with an LPAR of LP1, which has SYS1 as the system defined to run in it. Because you decided to use the Processor Operations interface protocol for PROC1, you can use the Processor Operations policy of your SYS1 system entry to specify basic LOAD information to use when initializing SYS1 under Processor Operations control. For more information about the different processor hardware interfaces, see “Operating the Processor Hardware Interfaces” on page 116.

To learn more about this entry type, see section "Processor Entry Type" in *IBM Z System Automation Defining Automation Policy*.

System (SYS)

This entry contains all definitions that System Automation needs to manage a system in the context of application automation. For Processor Operations use, the processor operations definitions are provided.

To learn more about the Processor Operations specific system policy definitions, see section "System Entry Type" in *IBM Z System Automation Defining Automation Policy*.

Enterprise (ENT)

For Processor Operations only, the PROCOPS FOCAL POINTS policy of your Enterprise entry must be defined. The minimum required definition is the primary focal point name. This is the NetView domain name of your System Automation Processor Operations instance. The backup focal point name specification is not used. It is reserved for future use.

For more information about this policy, see section "PROCOPS FOCAL POINTS Policy Items" in *IBM Z System Automation Defining Automation Policy*.

Mapping Processor Hardware Interface Items to Customization Dialog

Table 15 on page 113 lists the functional elements of the processor hardware interfaces shown in “Connectivity and Functionality at a Glance” on page 108 and shows their relationship to the related policy entries in the System Automation Customization Dialog.

Not all required data input fields of the listed policy entries are referenced in this table. To learn more about these policy entries and all required customization input data, see "Entry Type Reference", sections "Processor Entry Type", "System Entry Type" in *IBM Z System Automation Defining Automation Policy*.

Some policy entry customization items must also be defined in the SE or HMC configuration settings. Make sure that these definitions are identical in their readable form. An example of such items is the Auth Token, which is needed for an INTERNAL connection definition and must be defined in uppercase. See “Hardware Preparation” on page 111 for more information.

Table 15. Processor Hardware Interface Items

Processor Hardware Interface Items	Policy Entry	Related Policy Information and Comments
z15	PRO (Processors)	Policy: Processor Information Fields: Processor Type, Processor Mode, HW Resource Name, Network Name No processor hardware-specific name or device type must be defined. Use the generic type name `Mainframe` to classify the processor as real hardware. By default, the Mode is set to LPAR. This is correct for all supported mainframe processors. Your mainframe processor must have an address in the (SNA) format: Network_Name.NAU. The IBM mainframes have a factory set network name of IBM390PS and a NAU name that is derived from the unique CPC serial number. Both names can be changed by the customer. The SNA format processor address is stored in the SE and can be determined with the SE or HMC (CPC details).

Table 15. Processor Hardware Interface Items (continued)

Processor Hardware Interface Items	Policy Entry	Related Policy Information and Comments
CPC Support Element	PRO (Processors)	Policy: Processor Information Field: TCP/IP Address/Hostname or identifier ISQET32 for BCPii redirection Make sure that the address is valid in your customer network and that the host name gets correctly resolved.
HMC1, HMC2	PRO (Processors)	Policy: Processor Information Field: Alternate Address/Hostname or identifier ISQET32 for BCPii redirection Make sure that the address is valid in your customer network and that the host name gets correctly resolved. If you have multiple HMCs in your environment, make sure that the HMC that you plan to use for Processor Operations has your processor that is defined in its CPC group.
SNMP Path	PRO (Processors)	Policy: Processor Information Field: Connection Protocol The SNMP value enables Processor Operations SNMP communication.
TCP/IP Path	PRO (Processors)	Policy: Processor Information Field: Processor Type Use the PSM processor type to define a connection to a Processor Operations Service Machine, which is required to manage z/VM guests. This automatically assigns the connection protocol TCPIP for this processor.
INTERNAL Path	PRO (Processors)	Policy: Processor Information Field: Connection Protocol The value INTERNAL enables the BCPii connection protocol.
Logical Partitions (LPARs)	PRO (Processors)	Policy: LPARs and Systems Fields: LPAR Name, Target Mode, Target System Name Processor Operations requires that you define at least one LPAR of your CPC and assign a defined system to it. Otherwise, the interfaces do not work.
PSM	PRO (Processors)	Policy: Processor Information Field: Processor Type For the z/VM guest support, define a processor of type PSM. See “Other functions” on page 117 for details. Policy: Guests and Systems Here you define the name of the virtual machine as defined on the z/VM host system, the architecture mode of the virtual machine, and the corresponding target system name.

Table 15. Processor Hardware Interface Items (continued)

Processor Hardware Interface Items	Policy Entry	Related Policy Information and Comments
z/OS A, z/OS B, z/VM A	SYS (Systems) and PRO (Processors)	Policy: Systems > System Information Fields: Operating system, Image/System name For a list of the supported operating system environments, see section "Other functions" on page 117 for details. The Image/System name is used by Processor Operations as the target system name parameter to address commands to this system's CPC, LPAR, or integrated console. In LPAR Management commands, the target system name parameter is used as a shortcut of the parameter format *processorname.LPARname*. Policy: Processors > LPARs and Systems If you have not already assigned a system in the LPARs and Systems policy of the processor entry, link the target system to the LPAR that you want to use for this system. Unassigned target systems are ignored by Processor Operations. You can define a system entry for any of the supported operating systems. See "Other functions" on page 117 for details.
z/VM guest	SYS (Systems)	Policy: System information Fields: Operating system, Image/System name The z/OS, z/VM, Linux for System z, and z/VSE operating systems are supported by Processor Operations as z/VM guest operating systems. The name that is defined as the Image/System name is identical to the operating system name, which might be different to the system entry name. The Image/System name is used by Processor Operations as the target system name parameter to address commands to this guest system's PSM, virtual machine, or console.
Usage Examples	SYS (Systems)	Policy: Target System Info, IPL Info In the Target System Info policy, you can specify if Processor Operations should start a connection to the processor hardware/LPAR of this target system automatically, after the Processor Operations initial startup is complete. In the IPL Info policy, you can specify for z/OS or z/VM target systems the automatic responses to important IPL operator prompt messages.

Usage and Operation

Operating the Processor Hardware Interfaces

The System Automation product offers two hardware interfaces. Depending on your processor policy protocol definitions, one or both hardware interfaces can be used after the System Automation agent is started.

INTERNAL path

This interface is based on the internal BCPii processor communication protocol and does not require TCP/IP or Processor Operations to be active. No start or stop commands are available to control the interface. It is activated automatically as part of the System Automation start. A session to the Support Element of your CPC is established after the BCPii start is complete. When the System Automation agent is stopped, the BCPii connection to the CPC is stopped.

The INTERNAL path is exclusively used by GDPS.

Processor Operations – SNMP path

This interface requires TCP/IP, or can be redirected to INTERNAL path by defining a hostname of ISQET32. The z/VM guest systems can be also controlled.

Command	Description
ISQSTART	This command starts Processor Operations. Depending on your Processor Operations policy definitions for each entity, a Processor Operations connection is established to the target systems, CPC, and HMC.
ISQSTOP	This command stops Processor Operations. All active connections to the CPC, HMC, or z/VM PSM are closed.

Operation

Command	Description
ISQXDST	Processor Operations Status System dialog. Shows the status of the monitored components: processor (THW), system (TGT), and the communication link.
ISQCCMD	Interface to perform HW common commands, such as ACTIVATE, DEACTIVATE, or LOAD.
ISQSEND	The command communicates with the "Operating System Messages" integrated console (SE/HMC) for each LPAR, in order to send commands to and receive responses from the operating system running in this LPAR.

Details about the hardware interface commands can be found in "Processor Operations Commands" in *IBM Z System Automation Operator's Commands*.

HMC task mapping

Table 16 on page 116 shows the mapping of operator HMC tasks to the corresponding commands for each processor hardware interface.

Table 16. HMC Operations Management Tasks

HMC Operations Management Tasks	Processor Operations Using either TCP/IP SNMP or ISQET32
Load (IPL) a system	ISQCCMD LOAD
Power-ON CPC, activate LPAR	ISQCCMD ACTIVATE

Table 16. HMC Operations Management Tasks (continued)	
Deactivate LPAR, Power-OFF CPC	ISQCCMD DEACTIVATE
Manage CPC capacity	ISQCCMD TCM,CBU,OOCOD
Manage server time network	ISQCCMD STP
Manage CPC power mode	ISQCCMD POWERMOD
Manage hardware Activation Profiles	ISQCCMD PROFILE
Query CPC/LPAR configuration and status	ISQCCMD GETSINFO, GETIINFO, CPCDATA
Change LPAR settings	ISQCCMD ICNTL
Query CPC capacity, time protocol	ISQCCMD TCDATA, STPDATA
Query LPAR status data (for example, PSW)	ISQCCMD GETIPSW

Other functions

Operating system environments

While these systems are running on your processors, you can define them in the System Automation PDB as Processor Operations target systems. You can then perform base operator tasks, such as console message monitoring, LPAR management, activation, and IPLs for all these systems from a single point of control, the System Automation Processor Operations focal point. The following operating system environments are supported:

- Operating Systems: z/OS, z/VM, Linux on System z, z/VSE
- Coupling Facility Control Code (CFCC)
- Secure Service Container (SSC)

Managing z/VM guest systems with Processor Operations

If a z/VM guest system is running in an LPAR of your processor, Processor Operations can be used to manage guest systems running on this z/VM host. In your automation policy, define a PSM processor and define each guest that you want to manage as a system associated with the PSM processor. You can then use Processor Operations commands to monitor and control these guests.

To learn more about the policy entries, see section "Policy Items for PSM Processors" in *IBM Z System Automation Defining Automation Policy*.

For information on how to prepare and install the PSM support, see section "Preparing the VM PSM" in *IBM Z System Automation Planning and Installation*.

Writing automation scripts

All operator commands of the processor hardware interfaces are implemented as APIs. You can use these APIs to write your own automation REXX scripts, supplementing the existing set of commands with your environment-specific sequence of Processor Operations commands. For more information, see "Automating Processor Operations-Controlled Resources" in *IBM Z System Automation Customizing and Programming*.

Automating messages

In addition to the product System Automation entries in the automation table, you can add your own entries with the INGMSGU member. This is also valid for messages that are coming from the processor hardware interface. Messages with the message IDs of ISQ900I, ISQ800I, and ISQ700I might be of

special interest. For more information about these messages, see *IBM Z System Automation Messages and Codes*.

Alert notifications for Processor Operations managed CPCs

Processor Operations receives hardware messages and status changes from the target hardware for connected systems. Some hardware messages and status changes might point to severe problems, which might affect the availability of the managed systems. Processor Operations uses the INGALERT command to inject alerts into a system and provides built-in alert points for selected hardware problems. The following set of built-in alert points is available for Processor Operations:

Alert ID	Description	Resource Type
ISQHWMSG	One of the following important hardware messages was issued on the target hardware: • Hardware problem • Power problem • Licensed internal code problem • Licensed internal code detected a problem	MVSESA
ISQHWST	Target hardware (CPC) status that is changed to one of the following statuses: • DEGRADED • SERVICE REQUIRED	MVSESA

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

For Hardware Messages details, refer to the Hardware Messages task, which is available on the Hardware Management Console. For a list of the available CPC statuses, use the Acceptable Status tag in the details window that is shown for a selected CPC on your Hardware Management Console.

Chapter 8. Support

The following support procedures are available for managing your SA z/OS environment.

Modifying the maximum number of Language Processor (REXX) environments for NetView

The SA z/OS runs REXX code. REXX Language Processor environments are required to do so and are allocated on a system-wide basis. You might have to allocate more environments. Refer to *IBM Z NetView Installation: Configuring Additional Components* for more information.

Modifying the NetView subsystem interface procedure

The NetView subsystem is already created for you by the Configuration Assistant. However, if you have to set the parameters for the Program-to-Program interface (PPI) or set a particular command designator, see the following description.

Update the NetView subsystem interface procedure, CNMPSSI (CNMSJ010) member in the SYS1.PROCLIB data set, as required for your environment:

- "Changing symbolic variables" on page 119
- "Specifying the subsystem command designator" on page 120

Changing symbolic variables

Review and modify the following symbolic variables as required for your installation:

ARM
Enables the NetView subsystem for MVS automatic restart management (ARM) and supplies a name (up to 16 characters) by which this NetView subsystem is to be known to ARM.

The CNMPSSI ARM parameter can have the following values:

***ARM**
Specifies that the NetView subsystem is to be enabled. The NetView component generates a NetView ARM element name.

name
Specifies that the NetView subsystem is to be enabled and identifies the NetView ARM element name. This name can be 1 - 16 alphanumeric characters in length. The first character must be alphabetic. The name can contain the special characters #, @, and $.

***NOARM**
Specifies that the NetView subsystem is not to be enabled. *NOARM is the default value.

PPIOPT
Specifies whether you want to initialize the PPI facility. The PPIOPT parameter can have the following values:

PPI
Specifies that the PPI facility is to be started and initialized for the NetView subsystem address space. You cannot request this option for more than one subsystem address space. If the PPI is already active on another subsystem address space, it does not initialize for more requests. PPI is the default value.

NOPPI
Specifies that the PPI facility is not initialized for the subsystem address space.

P256BUF
Specifies the number of 256-byte PPI buffers. The default value is 300.

P4000BUF
Specifies the number of 4000-byte PPI buffers. The default value is 0.

REG
Specifies the region size for the NetView subsystem buffer address space in KB (the number specified times 1024). The default value is 16600.

The default value allows for 4200 total message and command buffers. To calculate the correct region size for your network, refer to *IBM Z NetView Tuning Guide*.

ROUTECDE
Specifies the route code to be used for messages that are issued by the SSI address space.

Messages that can be issued before this parameter is processed use the default route code 1, regardless of the value set here. Valid values are in the range 1 – 128.

&SQ1
Specifies the high-level qualifiers for user-defined data sets. The default value is `NETVIEW.V6R2M1`.

You can also adjust the CNMPSSI parameters using the SSI statements in the CNMSTYLE member.

Specifying the subsystem command designator

To specify the NetView subsystem command designator, use the MVSPARM.Cmd.Designator statement in the CNMSTYLE member. For more information about the MVSPARM.Cmd.Designator statement, see the *IBM Z NetView Administration Reference*.

Saving Automation Manager data sets into Generation Data Groups

About this task

During the lifecycle of an Automation Manager, log data is written into unique Automation Manager data sets. If an unexpected Automation Manager abend occurs, IBM service might request these data sets for problem determination. The best practice *BASE sample policy ensures that two Automation Managers are active at the same time. When an Automation Manager abend occurs, the Automation Manager is restarted by SA z/OS. Now, the content of the Automation Manager data sets is overwritten.

To prevent this situation, follow these steps to save the data sets into generation data groups before the restart of the Automation Manager instance.

Procedure

1. In the SA z/OS Automation Manager start procedure:
   a) Modify the startup procedure of your Automation Manager generated by the Configuration Assistant.

      Refer to the setting made in the `INGDOPT Configuration Options file` keyword `<sa_am_start_proc>`.
   b) Activate by uncommenting the SLQAUX symbol.
   c) Remove the job end statement.

      For more information, see the corresponding comment section in the JCL.
   d) Save your changes.
2. In the Automation Policy:
   a) Add the string `SLQAUX=AM1` in the **SYMBOL 1** field for the APL entry AM in policy APPLICATION SYMBOLS.
   b) Add the string `SLQAUX=AM2` in the **SYMBOL 1** field for the APL entry AM2 in policy APPLICATION SYMBOLS.
   c) Add the string `&SUBSSYMBOL1` in the **Startup Parameters** field for the APL entry C_AM in policy STARTUP.

d) Build the SOCNTL file.

e) Refresh the configuration in your runtime environment.

Additional configuration considerations

This section describes additional configuration considerations for setting up NetView components, including the following topics:

Configuring the operator environment

The following topics describe aspects of the operator environment that you can customize:

Defining NetView operators

You can define your NetView operators either by using an SAF security product, through DSIPARM member DSIOPF, or both. For detailed information about defining NetView operators, refer to the *IBM Z NetView Security Reference*.

Specifying the degree of security verification

The recommended degree of security for SA z/OS is already described in Chapter 4, "Security," on page 31. If you cannot follow these recommendations, read this section.

You can define the degree of security verification to be performed when an operator logs on by using the SECOPTS statements in the CNMSTUSR or CxxSTGEN member. For information about changing CNMSTYLE statements, see *IBM Z NetView Installation: Getting Started*.

Use the REFRESH command to refresh many types of security that is used while the NetView component is running. The REFRESH command can be used to change the security settings in the CNMSTUSR or CxxSTGEN member.

If you want information about...	Refer to...
changing CNMSTYLE statements	*IBM Z NetView Installation: Getting Started*
SAF checking	*IBM Z NetView Security Reference*
REFRESH command	*IBM Z NetView Command Reference Volume 1 (A-N)*

Defining PA and PF keys

During logon, an operator runs the PFKDEF command list, CNME1010, which (as a default) references keys defined in the sample CNMKEYS. This command can also be included in the operator profile.

To change the NetView default PF key settings or the default line of text at the bottom of many NetView panels that describes PF key settings, modify CNMKEYS.

For specific information about modifying CNMKEYS, refer to the *IBM Z NetView Customization Guide*.

Working with data logs and databases

Printing the Network Log

The network log is used as a protocol for all processes and actions that are completed within SA z/OS. This log also contains the diagnostics data in case of an error. IBM service might require this data. There are two NetView log data sets in place that are used alternatively by the DSILOG task.

You can use the INGEPRT JCL to print the NetView log. The INGEPRT JCL was copied to your PROCLIB data set during configuration. Further details are described in the INGEPRT JCL.

It might be advisable to switch the NetView log data set right before rerunning a scenario to gather service data. This switch might avoid alternating the NetView log while log data is gathered.

You can use the SWITCH DSILOG command to switch the NetView log. Refer to *IBM Z NetView Customization Guide*.

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.

Index

K

L

M

N

O

P

R

S

U

V

Product Number: 5698-SA4

SC27-9532-01